Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for informasjonssikkerhet og
kommunikajsonsteknologi

Sigurd Glende

Blockchain: Business Potentials

Masteroppgave i Kommunikasjonsteknologi
Veileder: Harald Øverby
Juni 2019

NTNU
Norwegian University of
Science and Technology

Sigurd Glende

Blockchain: Business Potentials

Masteroppgave i Kommunikasjonsteknologi
Veileder: Harald Øverby
Juni 2019

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for informasjonssikkerhet og kommunikajsonsteknologi

NTNU
Kunnskap for en bedre verden

Title: Blockchain: Business Potentials
Student: Sigurd Glende

Problem description:

The Blockchain industry is as of 2019 in a somewhat complex situation. Value of different players in the ecosystem has dropped significantly. While this does not have any direct indication of how development of the technology is proceeding, it could halt and restrain the progress. During the extreme growth seen in 2017, the eco-system sprawled with new ideas, companies, programmers and capital. The industry now faces considerable challenges regarding both technical and economic issues. As with most new technology, skepticism is present. Questions concerning price volatility, power consumption, scalability, security and consensus algorithms are looming large. Most developments are concerning said obstacles. There seems to be a race towards the "ultimate blockchain" with superior technology - completely undermining the business aspect.

In this thesis, the focus will be both on technological aspects, but most importantly, how this technology could be applied to business use cases. It is of high importance to look at the fundamental attributes of this invention, to understand where this technology can be applied - and actually create value for those adopting it. These fundamentals include immutability, finality and transactional history.During the research for this project, it has become evident that state of the art blockchain technology fits some sectors better than others. Players trying to apply these distributed ledgers to any problem will soon run into large obstacles. Supply chain management, with geographically disperse manufacturing sites and several players along the logistic network - all in need of information considering goods produced, shipped and sold, is a natural place to start. Such networks are complex, leading to problems such as lost information, lost products and especially pirated goods entering the supply chain.

The final product will utilize core aspects of blockchain technology. The main goal is to create a proof of concept demonstration of how blockchain technology can mitigate forged goods from entering supply networks. The system will consist of both a physical and digital part. The physical part will consist of scannable QR codes and RFID chips. The digital component will consist of a blockchain, smart contracts running on top of the blockchain, and a front-end that enable the user to interact with the ledger.

During the last semester, different blockchain systems were explored. Two candidates have been chosen based upon several factors regarding security, the technology, and most importantly how accessible the blockchain is. These two projects, named Ethereum and Vechain are becoming central players in the blockchain ecosystem. They both employ Solidity programming language, which will be used to create the smart contracts. The smart contract logic will define how interactions made by the user will alter states in the blockchain, such as information about geo-location, ownership and who is responsible for shipping. This will create data trails of origin and life-cycle for all products included in the system, greatly enhancing transparency and the ability to minimize forged goods.

Responsible professor: Harald Øverby, ITEM

Abstract

The environment revolving around blockchain has been heavily debated since Bitcoin's white paper was released in the wake of the financial crash of 2008. The idea of not having to rely on financial institutions to handle economic transactions was born. Since then, the life of the worlds first cryptocurrency has been turbulent. Volatile prices, extreme power consumption, congestion in the blockchain network and centralized, unstable exchanges are some obstacles that have to be overcome.

Since Bitcoins inception, the ecosystem have spawned thousands of blockchain building on Bitcoins underlying technology. Ethereum is one of the most interesting innovations. It is thoroughly examined and discussed throughout the thesis. Ethereum is often considered as "Blockchain 2.0", introducing smart contracts. These advanced blockchains can run what is named Decentralized Application (dApp). Immutability, finality and historical transaction data are core properties of blockchains, which are provided to applications built on them. These characteristics point us towards fields where blockchain technology might find a foothold and flourish. A field that fits these characteristics is supply chain management. Complex supply chains serve modern businesses with vast economic possibilities. However, they do come with an unfortunate quirk of allowing forged goods to enter these logistic networks. This raises questions about how we can control and survey complexity of global supply chains.

Blockchains provide a structure for storing open data in a secure matter. However, the mentioned obstacles hinder current blockchains of flourishing fully. The first part of this thesis - the background analysis, examines current blockchains and identify the most pressing issues. This part also focuses on the product counterfeit market, including major drivers, growth and consequences of the phenomenon. This part provide a satisfactory backdrop to understand how the two subjects can combine, and how complex supply chains with information asymmetry can be modified to combat the counterfeit market.

During the second part of this thesis, a proof of concept is designed and implemented. It details the design choices made such as the intertwined storage structure, utilizing both centralized and decentralized storage. The part outlines various technologies utilized to build and implement the application. Smart contracts enable business logic operations on the Ethereum platform. The decentralized application combines with QR-codes as a physical counterpart. The complete system enables verification of authenticity along the products' journey from inception to usage. Products such as luxury goods, vaccines and car parts are examples.

During the last part of this thesis, findings, limitations and the research questions are discussed. Some obstacles related to blockchain technology are more easily overcome than others. The findings show that latency appear as the bottleneck. Many approaches to mitigate issues compromises on the core fundamentals such as decentralization. Current blockchain structures are incapable of storing large amounts of data - a central issue that the proof of concept tries to mitigate by introducing centralized storage. Ensuring symmetric information and allowing those interacting in the system to audit open data is the chosen approach. This enhances supply chains resilience against counterfeit goods. However, there are several limitations regarding the system and the organizational culture within the field of supply chains.

Sammendrag

Den første kryptovalutaen, Bitcoin, oppstod i kjølvannet av finanskrisen i 2008. Ideen om å ikke være avhengig av store finansielle institusjoner for å håndtere valutatransaksjoner ble født. Siden den gang har et kjennetegn ved Bitcoin vært volatile priser, høyt energiforbruk og en overbelastning av nettverket. Alt kjennetegn som vitner om problemer ved den underliggende blokkjeden, problemer som må løses før teknologien kan brukes i stor skala.

Siden Bitcoins begynnelse har det blitt utviklet tusenvis av blokkjeder basert på Bitcoins underliggende teknologi. Ethereum er en av de mest interessante innovasjonene. Blokkjeden blir ofte betraktet som "versjon 2.0", og introduserer såkalte smartkontrakter. Avanserte blokkjeder gir støtte for såkalte desentraliserte applikasjoner. Uforandelighet og historisk transaksjonsdata er kjernegenskaper for de fleste blokkjeder, egenskaper som arves av applikasjoner bygget på dem. Disse egenskapene peker oss mot felt hvor teknologien kan finne fotfeste og blomstre. Et felt som passer disse egenskapene, er supply chain management. Komplekse verdikjeder gir store økonomiske muligheter, men introduserer også muligheter for forfalskning av produkter. Ut fra dette vokser det frem et spørsmål om hvordan vi kan kartlegge og kontrollere denne kompleksiteten.

Blokkjeder skaper en struktur for lagring av åpen data på en sikker måte. Nevnte hindringer begrenser blomstring av nåværende blokkjeder. Den første delen av denne oppgaven - bakgrunnsanalysen, undersøker ulike blokkkjeder og identifiserer de mest alvorlige hindringene. Denne delen fokuserer også på markedet for forfalskning av produkter, ser på dette markedets vekst, identifiserer drivere og beskriver konsekvenser. Denne delen gir et godt bakteppe for å forstå hvordan de to temaene kan kombineres, og hvordan komplekse forsyningskjeder med assymetrisk informasjon kan endres for å bekjempe forfalskning av produkter.

I del to av denne oppgaven er et konseptbevis utformet og implementert. Delen beskriver den sammenflettede lagringsstrukturen, som både bruker sentralisert og desentralisert lagring. Del to beskriver dessuten ulike teknologier som brukes til å bygge og implementere applikasjonen. Logikken i applikasjonen er utviklet ved hjelp av smartkontrakter på Ethereum-plattformen. Den desentraliserte applikasjonen kombineres med QR-koder som fysisk motpart. Systemet muliggjør autentisering av produkter i løpet av reisen fra produksjon til bruk. Produkter som luksusvarer, vaksiner og bildeler er gode eksempler.

I den siste delen av denne oppgaven diskuteres funnene, begrensningene og forskningsspørsmålene. Resultatene fra denne oppgaven viser at noen hindringer knyttet til blokkjedeteknologi er lettere løst enn andre. Responstid fremstår som flaskehalsen. Mange tilnærminger for å redusere problemene gjør ødeleggende kompromisser. Nåværende blokkjeder er ikke i stand til å lagre store mengder data - et problem konseptbeviset prøver å løse. Dette gjøres ved å introdusere sentraliserte databaser for lagring av data. Redusering av mengden asymmetrisk informasjon, og samtidig la aktører i systemet revidere åpen data er den valgte tilnærmingen for å gjøre verdikjeder mer motstandsdyktige mot forfalskede varer. Det er imidlertid flere begrensninger både med tanke på systemet og organisasjonskulturen i sektoren.

Preface

This master thesis has been submitted to fulfill the graduation requirements of the MSc in Communication Technology at the Norwegian University of Technology and Science. The thesis was conducted during the spring semester of 2019. It is based on a pre-study project conducted during the fall semester of 2018.

I would like to express my gratitude to my supervisor Harald Øverby for his guidance and reflections throughout the whole process, including the pre-project and the thesis. I would also like to thank my parents for supporting me during this period.

Contents

Appendices

List of Figures

Dictionary

Block Arrival Time The time elapsed between the arrival of two blocks.

Block Size The block size limits the amount of included transactions in a block.

Consensus Algorithm An algorithm that aid a blockchain network to reach consensus on transaction validity.

Ether The underlying token for transactions on Ethereum.

Hard Fork Radical protocol change which makes the blockchain *less* strict. Splits the blockchain in two.

Hashing Rate At what speed a blockchain network can compute the output of a hash function.

Layer-2 Off-chain solution.

Mining Pool Cryptocurrency miners pool their mining resource together to increase the chance of finding the next block.

Off-chain The movement of information and/or value not conducted on the blockchain..

Privacy Coin A cryptocurrency that enable privacy for identity (at minimum), transaction value and account balances..

Pure Function An Ethereum smart contract function that does not modify or read state variables.

Smart Contract Smart contracts are immutable contracts that reside on a blockchain. The lines of code which defines its logic are executed upon interaction.

Solidity The Ethereum programming language.

List of Acronyms

ABI Application Binary Interface.

API Application Programming Interface.

BTC Bitcoin.

DAG Directed Acyclic Graph.

DAO Decentralized Autonomous Organization.

dApp Decentralized Application.

dPoS Delegated Proof of Stake.

EUIPO European Union Intellectual Property Office.

EVM Ethereum Virtual Machine.

HTTP Hypertext Transfer Protocol.

IACC The International Anti-Corruption Conference.

ICO Initial Coin Offering.

IDE Integrated development environment.

IP Intellectual Property.

IPFS InterPlanetary File System.

JIT Just-in-Time.

JSON JavaScript Object Notation.

KLOC 1000 lines of code.

NPM Node Package Manager.

NTNU Norwegian University of Science and Technology.

OECD Organisation for Economic Co-operation and Development.

P2P Peer to Peer.

PoA Proof of Authority.

PoC Proof of Concept.

PoS Proof of Stake.

PoW Proof of Work.

QR Quick Response.

SC Smart Contract.

SEC U.S. Securities and Exchange Commission.

STO Security Token Offering.

TPS Transactions Per Second.

UTXO Unspent Transaction Output.

Chapter 1 Introduction

This chapter explains the motivation behind the project. The research questions are highlighted, and the most important contributions of the thesis are listed. Finally, the thesis structure is described.

1.1 Motivation

Utilizing blockchain technology is of increasing interest from several large companies around the world - even though we are yet to see a "break-through" application built on the blockchain. Due to its inherently low scalability, it is challenging to deploy most business applications on state of the art blockchains. Since the dawn of Bitcoin 10 years ago, the blockchain is still struggling to find a foothold, and identify which sectors suit the blockchain best. The world needs a better understanding of the blockchains fundamental attributes to fathom where it can be employed. To enable core properties to flourish, state of the art blockchains need alterations. Advancement barriers are thoroughly discussed throughout this thesis.

A central question whenever adopting new technology is how it creates value. When building blockchain dependant software, the application acquires important traits such as transparency, redundancy, security, public verifiability and data integrity. However, the compromises made to achieve decentralization reduce throughput, latency, privacy and enhance the cost. Blockchain technology is not a one-fits-all solution. Some use cases prosper with the introduction of the technology, while other software can become crippled by the compromises made.

A comprehensive research paper on supply chains claims that the two most pivotal barriers to reach effective supply chain management is "inter-firm rivalry" and "managerial complexity"[FMM08]. The study claim that the degree of resource sharing (e.g. information, knowledge) among partners correlates with the logistic networks' success. A mail survey from the same study reveals that the most significant barrier is inadequate information systems. Inappropriate or incompatible information systems reduce collaboration possibilities.

One of the most promising fields of use is supply management. The intricate networks of supply chain logistics are overdue for a simplification in the way information is stored and shared. Disentangling the current network can solve issues regarding lost information and forged merchandise entering supply chains. Blockchain technology is poised to enhance trust through transparency and traceability for exchanges of data, goods or any other financial resources. Several industry titans, such as Maersk, have expressed their interest in the technology [Gro17]. As with any new technology, it will take a vast amount of time, money and creativity for it to flourish into its envisioned role.

1.2 Research Questions

This project is a combination of two topics, blockchain technology and counterfeit products. To comprehend how the technology can be beneficial in combat with fake goods it is vital to understand the strengths and weaknesses of that technology. Research question one addresses these issues. Research question two applies blockchain technology to the topic of counterfeit products.

Research Question 1: What technical modifications are essential to make current blockchain solutions viable for businesses?

After conducting the pre-project, it became evident that current blockchains are not viable for most business purposes. Some of the identified obstacles are of such severity that business operations would become crippled if relying on a blockchain as an underlying technology. Research question two approaches these technological barriers and look at what needs to be accomplished.

Sub-question 1.1: What can be done to improve smart contract security? Smart contract security is of great concern. Once a smart contract is deployed on the blockchain, it can not be altered like regular code. Blockchains have thus enforced upon its user a new proverb: "code is law". When these smart contracts handle a vast amount of wealth, precise and secure coding is required.

Sub-question 1.2: How can scalability issues regarding storage, transaction speed (responsiveness), transactions per second (throughput) and transaction fees be conquered? Scalability is one of the major issues regarding state of the art blockchain technology. Where Bitcoin supports seven Transactions Per Second (TPS), Ethereum doubles the amount to 15 TPS. These numbers are inadequate for most purposes. Scalability has to be enhanced.

Sub-question 1.3: Which consensus reaching algorithms should be implemented? In Bitcoin, the consensus algorithm is called "proof of work". In addition to Proof of Work (PoW), there is a vast variety of other approaches to reach consensus.

These sub-questions help verify the examined blockchains adequacy. Understanding the fundamentals behind various blockchains enables a justifiable decision when choosing an underlying platform, and by extension greatly enhance the quality of the final product.

Research Question 2: How can a system based on blockchain technology diminish counterfeit products from entering supply chains and markets?

Having a back-end that is scalable, secure and transparent is crucial in the envisaged system. Scalability is needed to process a large number of transactions and data smoothly. Security, to handle vast amounts of wealth without setbacks. Transparency to facilitate information sharing between actors in the network.

1.3 Scope and Limitations

The scope of this thesis:

- Understand core components and limitations of the blockchain
- Create a comprehensive system based on blockchain technology for manufacturers and intermediaries in order to enhance transparency and information integrity throughout the supply chain
- Leveraging the created system to facilitate end user authentication

The first item relates to RQ1. The second and third relate to RQ2.

For tracking purposes, QR-codes are applied. The logic behind applying QR-codes in this proof of concept is mainly for convenience. QR-codes are cheap, easily acquired and easy to comprehend. RFID-stickers are suitable for a production version of this system, but are rejected as physical components for this PoC mainly because of complexity. Allocating time and resources to understand RFID technology thoroughly would reduce time spent acquiring and applying knowledge of other core aspects.

1.4 Thesis Structure

This thesis is divided into three parts.

Background and Related Work

Within this part, the background analysis resides. Chapter 2 provides an in-depth exploration of how the counterfeit product market is structured, why it has become a significant phenomenon, and how it affects actors across the globe. In chapter 3, blockchain technology is examined. Major projects are analyzed, as well as a comprehensive study of the underlying technology behind the innovation. The main obstacles going forward are delineated. Subquestions stated above (1-3) is referenced throughout this chapter. Finally, three projects that address the same problem as this thesis are reviewed in chapter 4.

Application Structure, Design and Implementation

First, this part defines the methodology and explain why the PoC approach was chosen. Chapter 6 describe how the application is structured and design choices made, as well as the technologies utilized. Chapter 7 explains the implementation and internal operations of the PoC.

Results, Discussion and Conclusion

The final part of this thesis examines results, discuss the findings and concludes on the research questions.

1.5 Contributions

The Decentralized Application

The main contribution is the decentralized application. The interface and storage structure can be regarded as separate contributions. However, the link between the client and the storage structure is also of importance. There are currently several projects that connect to a blockchain from a react application running on a desktop. There are, however, few projects that connect a react native application directly to an underlying blockchain.

Storage Structure

To overcome scalability issues that most blockchains face in 2019, off-chain storage is needed. The storage structure provided in this thesis is one approach to handle a growing amount of data.

The Code

The code is open source and public. Anyone anywhere can interact and analyze. As the technology is relatively young, learning how to connect components to the blockchain can be challenging. This code presents one approach.

Rundown of Blockchain Technology

For those new to the subject, it can be challenging to; 1. Get an overview of what the technology is and 2. understand details of the blockchains inner workings. This thesis provides a thorough examination of the blockchain ecosystem.

Part I

Background

Chapter 2 Product Counterfeit

This chapter presents section one of the background analysis. History and state of the art forged goods will be examined and explained. The complexity of the supply chain sector, allowing forged goods to exist will be detailed and analyzed.

2.1 Product Counterfeit

Since ancient times, product counterfeiting has been a known problem in markets worldwide. The phenomenon has existed for at least 2000 years - in an age when counterfeit coinage was a significant problem. As history shows, the trend is clear - wherever there are trademarks, counterfeiting follows [CZ13, p. 7-8]. With the current growing trade and complex globalization development, industries have turned their attention to "intellectual property" as a value generator [PS16, p. 11]. Actions infringing on such trademarks and other IP are punishable by law. Laws regarding such acts are advancing, but are outpaced by the market growth and an ever increasing product range. The consequences for communities affected by product counterfeit are extensive, including consumers dying of bogus pharmaceuticals. The world requires new, modernized counterfeit countermeasures to control the supply of forged goods better, and stall this trend.

2.1.1 Trademarks, Forgery and Laws

Trademarks are symbols used to identify goods and services with the manufacturer. Trademark counterfeiting refers to goods that are produced by one manufacturer, yet decorated with symbols or words of any other manufacturer.

In the US, the Trademark Counterfeiting Act of 1984(!) punishes intentional trafficking of counterfeit merchandise with up to five years of imprisonment, and/or a fine of up to 250,000 US dollars [Kea86, p. 121]. In nations such as France and Spain, where they have powerful brands - Hermès, Louis Vuitton and alike, the fines can range up to 300.000 euros, with a maximum of three years imprisonment [TCK16].

Even though laws regarding forgery are advanced in some countries, the majority look increasingly inadequate, considering the immense counterfeit market growth.

2.1.2 Market Size and Growth

Measuring the market size of counterfeit goods is not trivial. There are, however, some estimates that provide enough sources for them to be considered actual estimates, rather than guesses. Through a 2013 report, Peggy Chaudry (PhD) assisted by Alan Zimmerman, notes that the US government estimates a trademark counterfeit growth rate of 1700% over ten years from 1996 to 2006 [CZ13].

A report detailed by the The International Anti-Corruption Conference (IACC) in 2012 reports that "counterfeiting is a $600 billion a year problem". It is a problem that has grown over 10000% in the past two decades, partly fueled by consumer demand" [CZ13].

A report jointly conducted by the Organisation for Economic Co-operation and Development (OECD) and European Union Intellectual Property Office (EUIPO) estimates that global counterfeit and pirated goods were up to 2.5% of total global trade - amounting to a hefty 468 billion USD [PS16, p. 11]. Up to 116 billion of these are imports to the EU, accounting for 5 % of all EU imports.

The numbers are somewhat different from various sources. Although it is impossible to estimate the actual number, the market is undoubtedly immense.

2.1.3 Countries of Origin

The report done by OECD and EUIPO office suggests that any economy can trade with counterfeit products, either as the producer itself, or as a transit point in which forged goods flow through. Hong Kong, China and Turkey seem to be the essential players [PS16, p. 60]. Most emerging economies, such as those mentioned, along with others such as Greece, are subject to pirate trade. There are several reasons why this phenomenon has a foothold in these locations. Emerging economies often have sufficient infrastructure to ship manufactured goods, combined with soft institutional frameworks when dealing with counterfeit products. When these two traits are mixed, it creates a perfect foundation for illegal trade.

Because of complicated trade routes, where products are shipped via several transit points, it is hard to pinpoint precisely where counterfeit products are manufactured. These intricate global transport networks are one of the major drivers behind the extreme growth of illicit goods.

2.1.4 Major Drivers

There are many different drivers behind the immense growth of counterfeit during the last two decades. These drivers are important to analyze in order to fathom where a blockchain based solution can aid, service and counteract forgery.

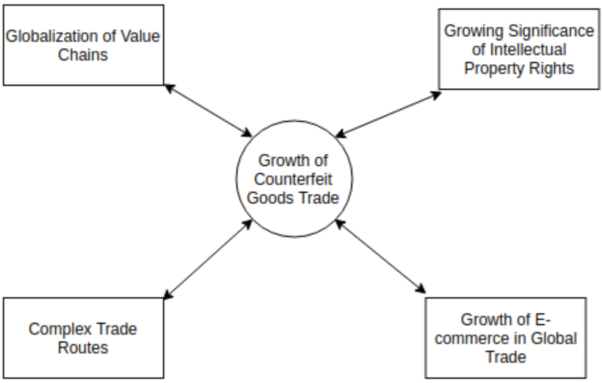

Globalization of Value Chains
Growing Significance of Intellectual Property Rights
Growth of Counterfeit Goods Trade
Complex Trade Routes
Growth of E-commerce in Global Trade

Figure 2.1: Four major drivers behind growth of counterfeit markets [CZ13].

Complex Trade Routes

The transit points mentioned above enable certain phenomenons to become part of the shipment routes.

- Camouflaging the actual point of product creation - "Origin Laundering."
- Free trade zones provide huge potential, as shipments of counterfeit goods can enter, and be divided into smaller shipments, creating an even more difficult exercise for those trying to prevent illegal trade.

- Repackaging goods at such transit points can provide counterfeiters opportunities to add counterfeit trademarks and re-label their shipments as they desire.

An example of this is a large number of counterfeit drugs that were revealed in Jebel Ali in Dubai, UAE. This seizure contained medications manufactured in China, shipped through Hong-Kong to the free-trade zone in Dubai. Then to Britain, then to the Bahamas, and finally back to Britain where the products were mailed to re-sellers with UK postage. These products were sold on an Internet site which made American customers believe they were buying medicines from a Canadian website [CZ13, 25].

Globalization of Value Chains

The globalization of value chains, combined with the post-2008-crisis recovery of global trade, creates a foundation for such transit points to become zones that enjoy an increasing amount of shipments. This revival of trade has handed previously untouched economies the ability to address a broader market. As mentioned, these emerging economies are subject to a high level of piracy trade.

The immense volume of imports in many countries, created by these globalized value chains, are of sizes that create an almost impossible job for Customs Services. "According to Deutsche Bank Research (2011) more than 25 million containers flowed through each of the ports of Shanghai and Singapore, ten million through Rotterdam and more than five million through Los Angeles in 2009"[CZ13, p.24]. The pace of growth has been extreme, World Container Traffic Data created by the International Association of Ports and Harbours show a ten-year growth rate of 137% in the number of containers flowing the top 20 ports. In Shanghai, this amounts to 37 million containers every year [oPH17].

Growing trade liberalization and enhanced access to new resources and markets create a geographically disperse manufacturing process. While the concept of outsourcing certain parts of the manufacturing process is well-known, recent technological developments have created possibilities of outsourcing to nations that were not available before.

Growth of E-commerce in Global Trade

As of 2019, any actor can connect to a wide variety of markets through the Internet. E-commerce introduces advantages for both sides of a trade. However, e-commerce has become a significant facilitator for counterfeit merchandise, enabling firms previously incapable of reaching a broad audience to sell their products to a global market.

Growing Significance of Intellectual Property Rights

Technically advanced products are rapidly becoming subject to counterfeit. The story of a fake apple-store in China is widely known [Tho11]. IP rights regarding such products are of great significance. A great deal of the value in these products are within intangible assets, the research and innovation behind the product.

Intellectual property for strong worldwide brands is a vital component that grants legal protection rights. Intellectual property rights provide a framework in which within they can legally enforce and prevent others from using innovation made by them - or in a modern approach, set up terms of agreements that allow them to trade intangible assets.

Unfortunately, it is much cheaper to steal than to innovate. When counterfeiters breach laws considering Intellectual Property (IP) rights, they attack innovation directly. There is no reason for large corporations to innovate if their intangible assets are stolen at launch. Protecting IP rights is of the utmost importance when fighting counterfeiters.

2.1.5 Expanding Product Range

The OECD has found a notable deviation from luxury products of high value towards conventional products - introducing counterfeiting to sectors previously untouched by piracy. Everything that is protected by either a patent, trademark or any other form of copyright, are subject to suffer from counterfeit. Even products like honey, cinnamon or coconut oil are subject to trademark infringement [Mol96].

Watches seem to be the product with the highest possibility of being pirated. Electrical machinery and equipment, clothing and leather comprise a significant part of the above 100 000 counterfeit seizures worldwide. Toys, footwear and pharmaceuticals are also commonly confiscated by customs. [PS16, p. 64]

The list is extensive - no product is safe from counterfeit in modern age. This is due to counterfeiters advancing technology, and their ability to deceive the customers, even when it comes to sophisticated products like cars and medicines.

2.1.6 Consequences of Product Counterfeit

Consequences of product counterfeit is a complex subject. Intellectual property right owners, wholesalers, nations and communities are all affected differently.

Firms and Organizations

As mentioned above, IP right owners are affected severely. Global powerful brands suffer various ways:

- Direct loss of sales
- Loss of goodwill
- Irreplaceable damage to corporate brand/reputation
- Trademark dilution
- Cost of protecting and enforcing their intellectual property rights

Nations

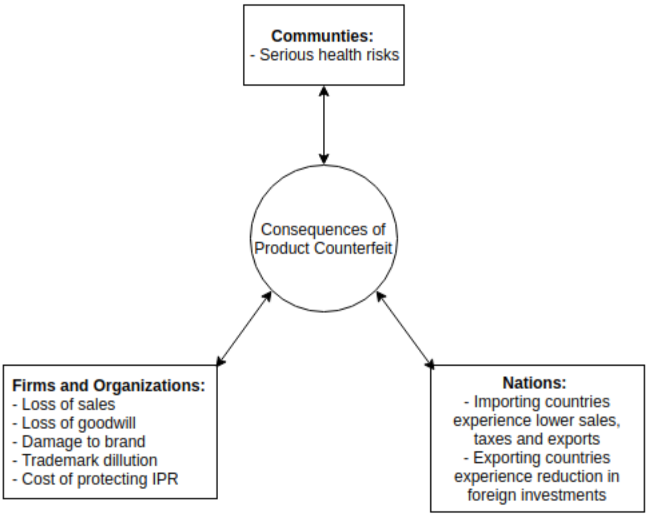
Firms in countries importing illicit goods can experience loss in sales, taxes and other revenues.

Nations exporting counterfeit goods can suffer from reductions in foreign investments made, due to the perception of the country being "an exporter of illicit goods" - firms might fear their IP being stolen.

Communities

The most obvious, and possibly most dangerous consequence of counterfeit goods affect the community. The consumers of products, whether it is toys, vaccines, phones or other items.

The process of producing medicines and vaccines is rigorous and expensive. There are vast amounts of research needed to innovate, create and build. The costly process correlates with the high price often found on such items. This is an opportunity for counterfeiters, enabling them to gain high margins on sales. Their products are usually not at the level required for pharmaceuticals, which can lead to severe complications for patients. During 2018 China faced criminals entering their pharmaceutical market, selling sub-standard vaccines to patients. President Xi Jinping describes the scandal as "vile and shocking" [Tim18].



Communties:
- Serious health risks
Consequences of Product Counterfeit
Firms and Organizations:
- Loss of sales
- Loss of goodwill
- Damage to brand
- Trademark dillution
- Cost of protecting IPR
Nations:
- Importing countries experience lower sales, taxes and exports
- Exporting countries experience reduction in foreign investments

Figure 2.2: Negative consequences of product counterfeit [CZ13]

2.1.7 Modernizing Counterfeit Countermeasures

There are undoubtedly several factors and concerns regarding product counterfeit. The laws are often too soft and can be difficult to apply. The market is growing, both in monetary value and geographical location, bringing new markets and nations into the problem. We have seen a growth in the importance of intellectual property and non-tangible assets, which drive counterfeiters further. Modernizing the governance of ever-expanding supply chains and trade routes can play a vital role in tackling this problem.

A 2014 study identifies specific promising countermeasures such as providing strict quantities of materials or intermediary goods in a JIT-structure [SB15]. Another countermeasure is sharing data with intelligence or customs.

As per today, traditional methods of fighting forgery are rapidly becoming ineffective. Consumers that want to acquire authentic products have to trust the seller of the objects - whether it is from a local store, web page or second-hand. Well-intended sellers have to trust transit intermediaries. Many products, such as Nike sneakers, Levi jeans or Omega watches, are all authenticated visually. Most consumers have no means to differentiate between legitimate and counterfeit goods. At every level throughout the supply chain, there is an inherent need to trust other actors in the network. The global trade network needs sophisticated methods of authentication that are difficult or unfeasible to work around for counterfeiters.

2.2 Supply Chain Complexity

The increasing complexity of supply chains is a significant concern. The complexity is caused by a collection of factors. Fernie and Sparks [FS18a, p. 33] believe that the four fundamental changes and challenges in retail logistics and supply chains are pace, span, availability and information. Emerging markets and the growth of already established markets are two central aspects regarding span and pace - the more objects being transported between additional locations, the more complex the network will get. Furthermore, we have seen the rise of a new concept called "fast fashion", which shortens demand response time drastically. The required pace of the supply chain is increased.

Market growth leads to growth in supplier and partner-relations. Managing these relations can become cumbersome. Different suppliers mean different inventories across multiple different locations. Traditional databases are "silo-based" - meaning that information residing in one organization or firm is not shareable. This mentality leads to asymmetric information and lowered supply chain visibility. The relationships are currently changing from the functional silos towards more collaborative relationships[FS18a, p. 53], a change which requires adequate underlying technology.

Customers demand more meta-data regarding the products they purchase, including supply sources and complete manufacturing history. Meeting these demands are often either too complicated, not cost effective, or even impossible given traditional supply chains[FS18b]. Data visibility needs to be further enhanced. The introduction of blockchain technology can aid traditional supply chains systems with immutability and transparency, whereas centralized systems can store large amounts of data.

Chapter 3
Blockchain

This chapter introduces the concepts of blockchains. Studying Bitcoin [Nak08] and Ethereum [B$^+$13] gives a broad understanding of most concepts applied in state of the art distributed ledger technology. This chapter addresses RQ1 with sub-questions: **What technical modifications are essential to make current blockchain solutions viable for businesses?**. This include hurdles related to consensus mechanisms, smart contract security, transaction speed, network scalability, privacy and price stability. Also, the most urgent regulatory and political problems related to blockchains are explored. Finally, different actors pursuing similar concepts to this project are delineated.

The subject of blockchains is broad. It includes cryptography, game-theory, hashing algorithms, momentum accounting and several other topics. Considering the breadth, providing a detailed explanation of the different areas would result in an undesirable length of this chapter. Thus, some sections are shortened down. For further reading material, *Bitcoin and Cryptocurrency Technologies*[NBF$^+$16] provide an in-depth look many of the underlying mechanisms.

3.1 Introduction

This introductory section swiftly examines Bitcoin and Ethereum. The two blockchains differ in many ways, even though they build upon the same fundamentals. Exploring their differences enlightens the fact that most blockchains are notably different, created to solve different tasks.

Blockchain has become a phenomenon during the last decade. The technology is encompassing. It is a bizarre melting point between technologies and professions. Experts in cybersecurity, mathematics and software development collaborate, create and innovate. While engineers try to tame the new technology, others have found great interest in blockchain technology in alternative ways. Analysts, traders, financial institutions, governments, regulators and alike try to understand this new technology. What is it? Where is it applicable? How should we regulate it?

The history of cryptocurrencies and blockchain is relatively short. Interestingly enough, the original Bitcoin white paper published in 2008 never mentions the term "blockchain". The technology has been dubbed so during the last ten years. Throughout the decade, blockchains have gone through several iterations and alterations. Two of the most important milestones are the dawn of blockchain, created by the unknown person(s) "Satoshi Nakamoto", and the creation of Ethereum.

3.1.1 Bitcoin

Bitcoin is a complex piece of technology. The network is created based upon ideas from several different publications, such as "b-money", an early proposal for an anonymous, distributed electronic cash system. Other ideas such as the 2002 "Hash-cash - a denial of service countermeasure" is also believed to have influenced Nakamoto[Nak08]. Even though not referenced, the system draws possibly the most influence from a proposal called "Bit gold", created by Nick Szabo in 2005[Sza05].

The Unspent Transactions System

One "bitcoin" defines as a chain of digital signatures. Users transacting on the network never really hold one or more bitcoins in their wallets. The account balance is an abstract notion. One users' balance is merely a representation of the value of all Unspent Transaction Output (UTXO) that has been sent to them, for which they hold the private key to verify further usage of this value. Those who own such UTXO's can transfer value from his address to another by signing a hash of the previous transaction in the chain. The sender of the transaction can verify the digital signatures to prove that he is, in fact, the owner of the coin.

One problem such systems can run into is a so-called "double-spend" problem. This event occurs when an owner spends one unspent transaction two or more times, and broadcast these to the network. Introducing a central trusted authority has been the favoured approach in many systems. This is where the Bitcoin protocol introduces its ingenuity by leveraging an open and universal ledger. The ledger records all transactions carried out in the network.

Maintaining Consensus

When a participant creates a transaction, it broadcasts its intention to transact value. Nodes in the network, called "miners" listen, and include this transaction in what is called "blocks". When enough transactions are included, and a proof-of-work puzzle is solved, one of these nodes broadcasts such a block. The block is then examined by *all* other miners, to ensure that they are valid. Once a majority of the miners accepts the block, it is verified, and the transaction is complete. This way, all verifying entities in the network know the entire transaction history. This comprehensiveness is termed "triple-entry bookkeeping", or "momentum accounting", a concept introduced as early as 1982 by an unfamiliar accounting researcher named Yuji Ijiri[IA82].

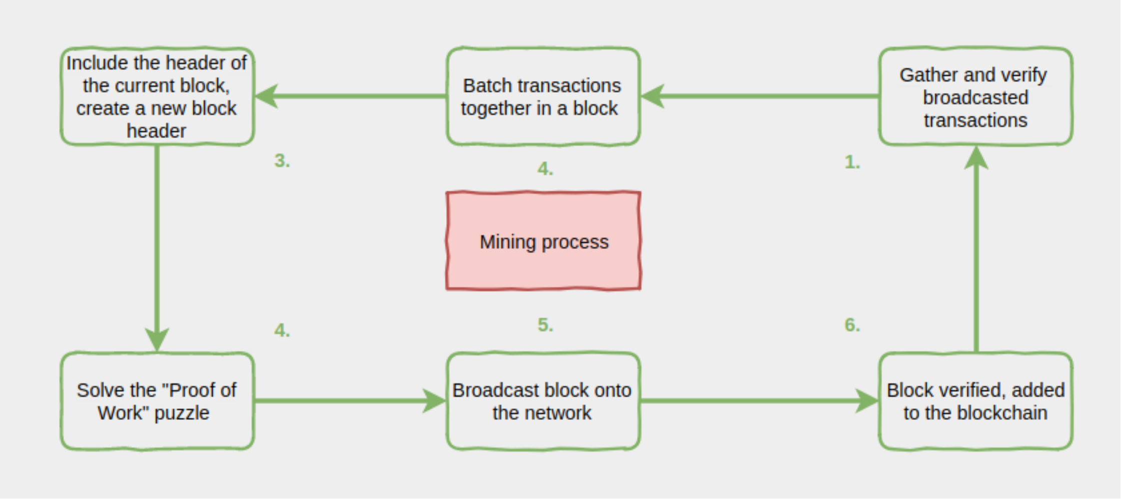

Include the header of the current block, create a new block header
Batch transactions together in a block
Gather and verify broadcasted transactions
3.
4.
1.
Mining process
4.
5.
6.
Solve the "Proof of Work" puzzle
Broadcast block onto the network
Block verified, added to the blockchain

Figure 3.1: The proof-of-work mining process in Bitcoin

Bitcoin introduces a concept called "Proof of Work" to ensure that all miners act accordingly. Before a miner broadcasts a new block, it has to solve an increasingly hard puzzle. When solving this puzzle, the miner allocates computing power and electricity, thus having a stake in the network. If the broadcasted block is verified, the miner gets his reward in the form of bitcoins. The monetary reward is what keeps the blockchain network running - the miners are incentivized to operate according to the specifications.

Proof of Work is one of many ways to keep consensus in a blockchain network. This mechanism has stood the test of time, running for over a decade without any major hiccups. One criticism of this system is its enormous power consumption. Other mechanisms such as "Proof of Stake", "Delegated Proof of Stake" and "Proof of Authority" have since been introduced. They all have advantages and disadvantages, which are examined in subsection 3.2.2.

3.1.2 Ethereum

Ethereum builds upon many concepts adapted from Bitcoin. By introducing smart contracts, Ethereum greatly enhances the capabilities of a blockchain. Through employing the native programming language "Solidity", developers can create decentralized applications that run on top of the Ethereum network. Ethereum is often termed as the "second generation of blockchains".

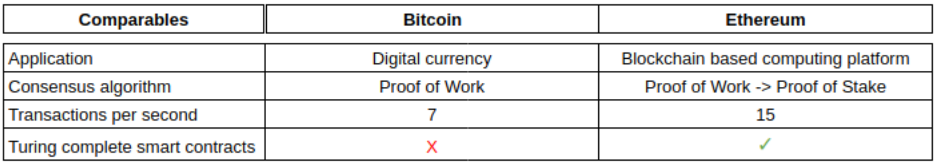
World State

In contrast to Bitcoin, there are actual account balances in the Ethereum network, managed by the "Ethereum world state". Activities on the network, such as transacting value, mining or creating smart contracts change the state of the Ethereum blockchain. The data is hashed and saved in blocks. Roughly every 15 seconds the Ethereum blockchain verifies a new block. Every block is a snapshot of how the world state was at that exact moment in time, thus creating a history that is auditable for miners in the network.

To maintain consensus of the world state, miners creating new blocks utilize the PoW-concept introduced by Bitcoin. As mentioned, this system uses extreme amounts of energy in order to function. Another drawback of state of the art PoW is its inability to scale. Where Bitcoin blocks have an arrival time of ten minutes, Ethereum has the mentioned 15 seconds - enhancing both transaction speed and throughput, but it is still not adequate. The system has stalled several times since its inception, due to the sheer amount of activity running through dApps implemented on the network. The Ethereum Foundation, which is in charge of research and development for bettering the Ethereum Blockchain, has decided to switch its consensus algorithm to proof of stake. This will decrease power consumption tremendously and enhance scalability. As on 2nd May 2019, the PoS transition seems close, with an expected code implementation before the end of June 2019.

3.1.3 Blockchains Are Not Equal

Even though built upon many of the same concepts, blockchains are not equivalent, and should not be treated alike. One important distinction is the differences between a "cryptocurrency" and a Blockchain platform. A cryptocurrency is a digital currency that can be used to transact value between two parties, peer-to-peer without intermediaries. A Blockchain is an incorruptible digital ledger of economic transactions and is often the underlying technology of cryptocurrencies. The platform is often Turing complete and can solve complex computations.



Comparables	Bitcoin	Ethereum
Application	Digital currency	Blockchain based computing platform
Consensus algorithm	Proof of Work	Proof of Work -> Proof of Stake
Transactions per second	7	15
Turing complete smart contracts	X	✓

Figure 3.2: Two blockchains, serving completely different purposes

Ethereums native currency "Ether" is the value transacted on the Ethereum network, and while it can be used to pay for services and goods (such as the intended use of Bitcoins), that is not its intended purpose.

3.2 Technological Overview

Keeping in mind that all blockchains differ, aspects inspected in this section are those most widely used. The network architecture consists of transacting parties and nodes governing the system. Every ledger needs to preserve consensus between these nodes to keep it running. When consensus is preserved, smart contracts can run continuously on the network, allowing execution of complex computations.

3.2.1 Network Architecture

There are several different approaches when it comes to building the network architecture supporting a blockchain platform. Blockchains can be divided into three different types. Private, consortium-based and public blockchains.

Private Blockchain

Private blockchains are blockchains where write and read permissions are kept centralized to one entity or organization. The participants in the network are well known. Private blockchains are often deployed on a single machine, whether it is a physical or a virtual one. Private blockchains include many traits coinciding with a traditional database. For this project, a completely private blockchain would be disadvantageous due to the need for a robust network without single points of failure. A private blockchain may be regarded as "a traditional **centralized** system, with a degree of cryptographic auditability" [But15].

Consortium Blockchain

A consortium blockchain is situated between two categories, private and public. This type of blockchain is a ledger where a number of selected nodes preserve consensus. The participants in the consortium are different organizations, that would like to collaborate and enhance data transparency between parties. The right to read and write on the blockchain can be public or restricted to those who compose the consortium. Hybrid versions of the two are also possible, where "the public" can verify that the consortium is operating legally in regards to a set of pre-structured rules. For example, a set of 10 financial institutions might utilize such a consortium blockchain solution. "The public" in this illustration could be third parties who audit the network to ensure that all participants are treated correctly. These blockchains can be regarded as **partially decentralized**.

Public Blockchain

A public blockchain is a blockchain on which any entity in the world can read and write. Examples of such are the two blockchains compared above, Ethereum and Bitcoin. On these, anyone in the world can transact with each other. The set of nodes governing the system is no longer pre-selected - those who see an incentive to join the governing process are able to. Combining economic incentives and cryptographic verification using consensus algorithms such as PoW or PoS help the system stay in harmony and consent. These systems can be regarded as **fully decentralized**.

3.2.2 Governance - The Consensus Algorithm

This section examines RQ1.3: **Which consensus reaching algorithms should be implemented?**. The governing mechanism of any distributed computing or multi-agent systems is vital. It is the core structure that aids nodes in the system to reach consensus. Keeping consensus is complex, especially in vast public blockchain networks. The most important job of a consensus algorithm is ensuring that the next block in a blockchain is the only version of truth. If implemented correctly, consensus algorithms ensure that governing nodes are mining blocks correctly, even in trustless ecosystems. How blockchains reach consensus differs greatly.

Proof-of-work

Proof-of-Work, PoW, is the oldest and "original" consensus algorithm in the blockchain space. First utilized in Bitcoin, it has since been used by numerous other large blockchains. As mentioned earlier, miners in the network compete against each other to compute the next valid block and obtain the reward.

Pros:

- Thoroughly tested, running for over a decade
- Widely adopted, well understood
- Easy to integrate new validators

Cons:

- Extreme power consumption
- Vulnerable to 51% attacks, where an adversary gains a majority of the hashing rate in the network

There are several other versions constructed upon principles of PoW, such as Proof of Elapsed Time and Proof of Activity.

Proof of Stake

Proof of Stake (PoS), is considered one of the most promising consensus algorithms. This mechanism chooses a miner of a new block, or in this case often called a "validator", in a semi-random process. Validators of blocks put value (in the form of "deposits") at stake and are rewarded with the ability to validate blocks. Higher stake equals a better chance of being chosen as a validator. It is essentially network security by putting economic value-at-loss. The penalties for a validator with malicious intent are many times larger than the amount earned by "staking" - in some cases the full deposit. The value obtained while staking is often a payout in the form of transaction fees.

In PoS, there is no battle to complete a puzzle in order to gain the right to produce a block. The producer of the next block is chosen based upon how large chunk of the underlying cryptocurrency the validator own, thus "semi-random". This consensus scheme greatly enhances scalability.

Pros:

- Negligible power consumption
- Improved scalability
- Not susceptible to 51% attacks

Cons:

- Not as thoroughly tested
- "Rich get richer" - only those meeting a certain set of prerequisites will be considered as validators

Delegated Proof of Stake

Delegated Proof of Stake (dPoS) is one version of PoS. In short, it is a form of digital democracy. It utilizes voting combined with a reputation-system to reach consensus. Token holders - those with a stake in the network, can exercise their right to influence decisions made in the network. The system consists of a set of "delegates", which are voted into their roles by the token holders. While this somewhat centralizes validation responsibilities to a smaller set of nodes, the whole network is effectively represented by these.

Pros:

- Democratic
- Highly scalable
- Negligible power consumption

Cons:

- Susceptible to 51% attacks
- "Rich get richer"
- Lack of "true" decentralization

Proof of Authority

Proof of Authority (PoA) is an alternative approach to reaching consensus. Explained best when compared to PoS. Instead of placing monetary value at stake, you put the reputation at stake. To achieve this, the true identity of validators is needed. Staking your true identity means disclosing who you are to earn the right to validate blocks.

Pros:

- Highly scalable
- Negligible power consumption

Cons:

- Only applicable in consortium blockchains
- "Rich get richer"
- Demanding process of choosing and maintaining validators

3.2.3 Smart Contracts

The contract is one of the most elemental components of a free market economy. Paper-based contracts have over centuries become binding agreements by law. Nick Szabo raised the question concerning how these could be converted to law-binding code in 1996 [Sza96]. He labelled them "smart" because they are considered far more functional than their paper-based counterparts. Szabo continues to explain that smart contracts reside on different every-day equipment such as vending machines and points of sale. The vending machine consumes coins and dispenses products - a simple contract between the payer and the machine. The problem with such contracts has been their inability to withstand attacks from adversaries. With the introduction of blockchain technology, smart contracts are now able to reside on top of this open source software without the possibility of modification or manipulation.

On Ethereum, these smart contracts are associated with an address. Interacting with this address equals interacting with the smart contract. The code is open source for anyone to audit, increasing the importance of precise coding. Smart contracts are in many ways structured as a class in object-oriented languages, featuring state variables and functions.

Smart contracts introduce an endless array of possibilities, but due to blockchain's immutability trait, a poorly written piece of code can result in the loss of vast amounts of monetary value. Due to this imminent danger of value loss, different ways of approaching the problem have been proposed. Managing critical smart contract code can be done in many ways, including through automated generation of smart contracts based upon a set of predefined rules, third-party auditing, or utilizing pre-programmed blocks to construct the contract. These are examined in subsection 3.3.3.

3.3 Technological Obstacles

There are several technological hurdles to jump through before blockchain technology can be considered as a viable solution in most business use-cases. As earlier mentioned blockchains are inherently slow. With Bitcoin boasting only seven tps, it cannot by itself aid the world of financial institutions. The scalability problem is arguably the largest, most pressing issue with blockchain technology as of 2019. Other issues include privacy for those utilizing a blockchain's services, smart contract security and cryptocurrencies' price volatility.

3.3.1 Scalability and Transaction Speed

During this section, RQ1.2 **How can we conquer scalability issues regarding storage, transaction speed and fees?** is addressed.

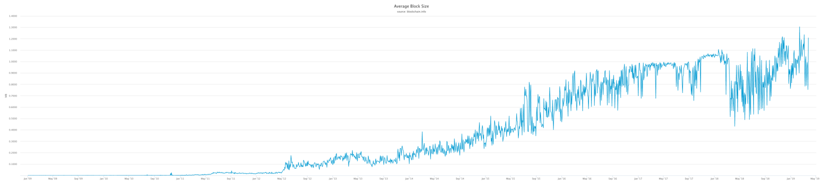


Figure 3.3: Average block size from 2009 until 2019. [Adapted from blockchain.com]

In the early days of Bitcoin, the network was quite idle compared to the present activity. Less network activity equals fewer transactions across the platform. With fewer transactions, there was no imminent issue regarding the scalability of the network. Indeed, most blocks before 2011 were in the size order of 1 kilobyte. Fast forward to 2017, the maximum block size set at 1 megabyte was under pressure (figure 3.3). There were too many pending transactions waiting to be included by the miners. Miners can choose which transactions they want to include in the next block. Transactions that pay high fees are attractive, which results in extreme transaction fees - as witnessed during the fall of 2017.

Block size and Arrival Time

Size and the arrival time of blocks are two parameters that can be changed to increase throughput and transaction speed. Larger block size results in more space for transactions in a single block. The current Bitcoin protocol specifies 10 minutes as the block arrival time. If that were to be shortened down to 5 minutes, the system would operate with twice the speed. There are, however, several problems with implementations like these. Such alterations provide temporary benefits only. As the network keeps expanding, the new cap will be met in time, causing the same problems as in 2017. Even if the cap increases to such an extent that there is no longer an issue regarding transaction speeds, other concerns arise.

At all times, there are nodes in the Bitcoin network that keep a complete log of all transactions that have occurred in the Bitcoin network, called full nodes. If the block sizes were to be increased by a hundred-fold to handle future activity, the size of the blockchain could grow 100 times faster than current growth. The complete transaction log would soon be deemed too large for most computers to handle. This would, in turn, centralize the operation of the network, since large cloud servers would be the only viable facilitator for full nodes. This phenomenon would also occur if arrival time were to be reduced.

Increasing the block size would also lead to problems regarding the broadcasting-mechanism that miners communicate through. Once a block is found by one miner,

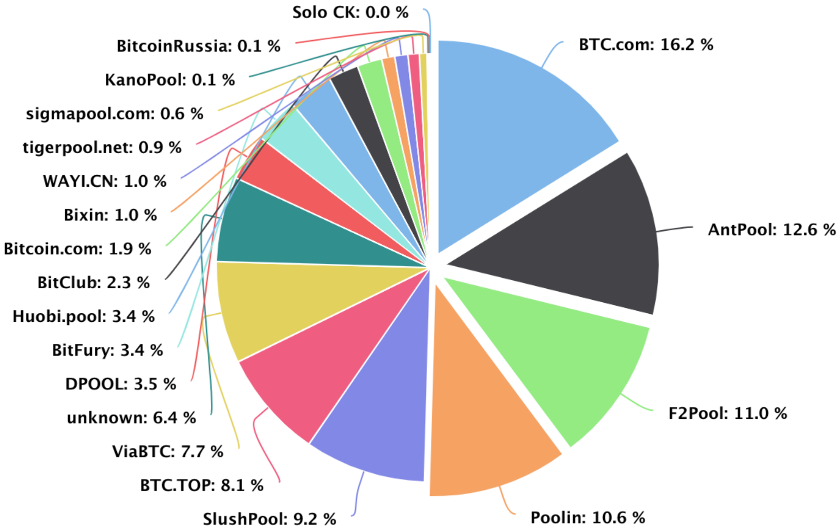


Figure 3.4: Bitcoin mining pool distribution during March 2019. Statistics gathered from BTC.com [BTC19].

it is broadcasted to all other miners. When the block reaches other miners, they are validated. Larger block size results in a slower broadcasting process, as well as a decrease in validation speeds. While all other miners are validating the broadcasted block, the broadcasting miner can get a head start on the next block. This will also result in centralization. The closer a miner is to the geographical location of the broadcasting miner, the faster it will receive the broadcasted block. This problem is already becoming significant in the Bitcoin network - as we have seen a trend towards large "mining pools" controlling most of the network. In reality, 50.4% (see figure 3.4) of the hash rate in the Bitcoin network is now controlled by the largest four mining pools (as for Ethereum the four largest control ~75%!). One could already imply that the mining process is centralized, as these four alone could crash the network.

The current implementation of PoW is cracking at the seams. Further centralization due to increased block size would not be advantageous for these blockchain platforms. Many solutions have been proposed. One of them is to create new blockchains such as Litecoin[Lee11]. Litecoin utilizes a different hashing-algorithm than Bitcoin, and have shorter block times. As earlier mentioned, the consequence is a faster-growing blockchain (with equal amounts of network activity) compared with Bitcoin. Once Litecoin reaches its limit, which it inevitably will if network activity increases enough, it will face the same issues as Bitcoin. Other distributed ledgers such as IOTA, NANO and Zilliqa experience much higher throughput. The first two mentioned projects utilize a Directed Acyclic Graph (DAG), a technology not thoroughly tested, and thus cannot be concluded as a complete solution to the scaling issue at this point. The latter utilizes a technology called sharding; a mechanism also implemented in traditional database systems.

The final solution might be two-fold. The first considered approach is through off-chain solutions, known in the community as "layer-2 scaling". This is an approach where some transactions are handled off the blockchain platform, only interacting with it sparsely. The second approach is to modify and alter the protocol design, such as the mentioned sharding.

Layer-2 Solutions

The two largest projects focusing on off-chain scaling are "the lightning network"[PD16], a project that centres around scaling the Bitcoin network. "Raiden Network" is an Ethereum based project engaging the scaling issue with the same approach as the LN-network[RN].

The idea is to set up a contract between payer and payee. On the Bitcoin network, this is done by creating a multi-signature address. The payer loads the address with value, e.g. 0.05 Bitcoin (BTC). This address can be regarded as a contract. The current contract is a "refund-contract", that sends 0.05 BTC back to the payer, and 0.00 BTC to the payee. Whenever the payer interacts with the payee, the contract is updated. Let's say the payer buys coffee from his local bistro for 0.01 BTC. The updated contract now states that the payer is to receive 0.04 BTC, the payee 0.01 BTC. This can be done repeatedly until the payer has spent his 0.05 BTC. The updated contract can be broadcasted to the network, and the payment is fulfilled. Both parties have identical copies of the contract, signed by both parties - but the transaction is not necessarily broadcasted onto the network. At all times, both parties can broadcast the contract and fulfill payments specified in the current version.

By utilizing multi-signature addresses in this fashion, payment channels can be set up between payer and payee. Two transactions are needed, one to set up the channel (the initial refund-contract), and another one to close the channel (the final contract between payer and payee). Those who require recurring payments are the primary targets for this network, given that it reduces a vast amount of transactions to two. The system Raiden Network is building is also supporting off-chain transactions with payment channels. These payment channels can be interlinked to create mesh-networks.

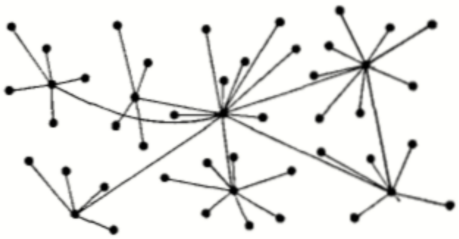


Figure 3.5: What the lightning network could look like - hub and spoke.

Payment mesh network

- Bob has a payment channel open to his local bistro.
- Alice, a friend of Bob, have a payment channel open with Bob.
- Alice now wants to buy coffee from Bob's local bistro.
- Alice pays the bistro through Bob, utilizing him as a middle man.

This mesh network only serves a real purpose for recurring payments. While this opens up possibilities of payment streams, such as real-time payments for broadband, or services supplied by, e.g. Netflix, it does not solve the scalability problem of Bitcoin. Criticisms also include the fact that such a mesh-network would centralize payment operation in the Bitcoin network since it reintroduces intermediaries. As depicted in figure 3.5, it could create large hubs that are interconnected with other large hubs, in order to involve every player in the network. These large hubs could introduce a fee for operating payment channels, effectively controlling the cost of payments through LN.

Sharding

Technically, sharding is a synonym for horizontal partitioning. In traditional databases, this technique is used to make large databases more manageable. When implemented, large databases become smaller, distributed entities. Examples are splitting a database geographically or logically.

The sharding technique could be applied in both processing and storage. This would allow miners to verify a subset of transactions made in the network, and full nodes to store a subset of the transaction history. Nodes only checking a subset of all transactions would greatly enhance transaction speeds in the network. Implementing sharding correctly to a blockchain is not an easy feat. Keeping all shards coordinated and withstanding "1 percent attacks" where an adversary controls one shard completely, are two problems at hand. Ethereum co-founder and CEO, Vitalik Buterin, claims to have created a sharding solution where both these problems are negated. While it is not yet implemented, sharding looks increasingly promising. As for Ethereum, sharding is a protocol level implementation, and will hence be "hidden" from developers, businesses and other end users.

While switching from PoW to PoS itself scales the Ethereum blockchain for the better, a correct implementation of sharding would complement this switch and further enhance scalability.

3.3.2 Privacy

The Importance of Privacy

Blockchain platforms like Ethereum bring an enormous amount of possibilities to the table. However, use cases can fall short of those requiring high levels of privacy, since blockchains such as Ethereum are completely transparent. Some actors in the network might want to transact without the possibility of being exposed. Examples of such actors might be large financial institutions interacting with customers. Data exposure is a real issue for institutions managing information such as bank accounts, social security numbers or intellectual property. The stated privacy issue is thoroughly discussed in the ecosystem. Several projects are pursuing a solution to the issue, with different approaches. Below are two interesting approaches.

Zether Smart Contracts

Zether is a project conducted by two Stanford students alongside two Visa Researchers. They have approached the problem by designing a new type of smart contracts. The new smart contract, called "Zether Smart Contract" includes a token called Zether Token, which is transferable between Zether accounts. It keeps account balances encrypted, but methods such as deposit, transfer and withdrawal of funds are exposed to its users through cryptographic proofs [BBB19]. These Zether contracts reside on the Ethereum blockchain.

Enigma

Another notable project is Enigma, a decentralized supercomputer built on blockchain that runs private computations [GZ15]. The enigma platform can take orders from Ethereum, bringing confidentiality and integrity through something dubbed "secret contracts".

When data is processed by nodes operating in the Enigma network, they do not have access to the complete set of data that resides in one smart contract, but rather a meaningless, random part of it. Only a small subset of all nodes are performing each computation over different parts of the data, somewhat similar to the mentioned Sharding scaling technique. This decreased redundancy dramatically enhances the scalability of the Enigma supercomputer. As an example withdrawn from the Enigma white paper states: "a group of people can provide access to their salary, and together compute the average wage of the group. Each participant learns their relative position in the group, but learns nothing more about other members' salaries." This gives us the possibility of securely evaluating computations without ever having to share the inputs.

3.3.3 Smart Contract Security

This section will address RQ1.1: **What can be done to improve smart contract security?** Next to scalability, smart contract security is one of the most important aspects of the blockchain. This is due to reasons mentioned earlier; open source code and large monetary value at stake. Even if the fundamental blockchain underlying these smart contracts is perfectly secure, a poorly written piece of code residing in an SC will eventually cause value loss for those having stakes in said contract.

The smart contract development community is on a mission towards zero bugs. David Gerard, the author of "Attack of the 50 foot Blockchain," claims that smart contracts are "a very bad idea in every way" [Ger17, p. 101]. He continues to explain that "computer code maps very badly to real-world legal agreements, where the hard part is not normal operations, but what to do when things go wrong; immutability means you can't fix problems, programmers need to write perfect bug-free programs first time every time, and the contract can't be updated if circumstances or laws change". These comments perfectly depict the "DAO" hack.

A Decentralized Autonomous Organization (DAO) consists of several smart contracts that are interlinked, forming an organization that is completely embedded into the code. The DAO-hack was likely due to poorly written code, which resulted in a ~50 million USD digital heist. These unfortunate events happened due to a recursive vulnerability bug, that according to its creators "did not put DAO funds at risk" [Tua16]. A "hard-fork" was since implemented to solve the issue and stop such a vulnerability being exploited elsewhere - revealing that this problem extended further than just poorly written smart contracts. The development of critical code should be slow and somewhat cumbersome. Ways of mitigating bugs and trying to achieve bug-free code are detailed below.

Mission for Zero Bugs

What this and several more events of similar character have shown, is that smart contract security is of the utmost importance. Solving this issue is not straight forward. Firstly, those writing contracts that can be classified as "critical code" should have the ability to do so. They should follow a specific set of rules as contracts are being developed, such as the "The Power of Ten – Rules for Developing Safety Critical Code" [Hol07]. These rules, developed by NASA, provide developers with direct, clear instructions on how to achieve high levels of code quality. The industry average, which is about 15 - 50 bugs per 1000 lines of code (KLOC) [Ste15], is not adequate for this industry.

Auditing

Thorough auditing by **several** trusted, relevant and experienced third parties is vital to reach desired levels of security. Since this auditing process requires such parties to be both competent in law and technically experienced, locating individuals possessing knowledge from both fields can be challenging.

Automated Generation of Smart Contracts

Auditing can be labeled as a "post-creation smart contract security measure". While this is a valuable step towards required security levels, it is not absolute. Concrete measures need to be in place not only before, but also throughout and after the creation process of these critical contracts. This introduces concepts such as semi- or fully-automated smart contract development. Fully automated smart contracts containing complex business logic might still be some years away - but not infeasible. A proposal has been made through "From Institutions to Code: Towards Automated Generation of Smart Contracts" [FN16]. This paper outlines an approach to translate human-readable contract representations into computational equivalents. The work conducted intends to bridge the gap between institutional specifications and machine-readable encoding in the form of Solidity contracts. They utilize a framework abbreviated ADICO to boost readiness for translation. The paper states that the generated contract skeletons "require considerable manual input to make them executable". While this solution is not ready, it can be with future efforts. A solution based on automation could provide several advantages.

- The threshold for developing smart contracts would greatly decrease since no code is touched by its creator
- Cost of creating correct code every time would be significantly reduced given that the code is re-used

Code made with such arrangements would mostly be day-to-day agreements. Such as the ownership transfer of a car or an agreement made between employee and employer.

Building Blocks

This concept includes utilizing predefined code snippets as building blocks. While somewhat similar to the idea above, this does not incorporate automation. Using building blocks is something with which most developers are familiar. In 2017, there were 350,000 Node Package Manager (NPM) packages available for developers in the JavaScript Node.js framework [BRO17]. The package registry has no vetting process for submission. The consequence is a higher degree of low-quality packages that are insecure or even malicious. A similar approach for the blockchain ecosystem would rely **heavily** on both third-parties and community experts to audit the packages. As an example, there are already some packages in usage on the Ethereum blockchain. "Safe Math" is one such package, created by OpenZeppelin, a framework of reusable smart contracts for Ethereum. Utilizing standard, tested, and community-reviewed code reduces the risk of vulnerabilities.

3.3.4 Price Stability

Transacting parties on the Ethereum network are required to pay GAS. Both for direct transactions between two individuals, or interactions made with a smart contract. The cost of transacting differs with the amount of computation needed to fulfill the transaction, and with the amount of data stored on the chain. Acquiring gas can be regarded as a method of reserving the right to utilize computational power and storage space within the Ethereum blockchain. GAS is also a crucial part of Ethereum's anti-denial of service model. If there were no fees, an adversary could easily halt the network with a DoS-attack.

During periods with large amounts of transactions, the cost of acquiring computational power increases as well - supply and demand. A typical computation on the Etherum network cost 1 Gwei (smallest unit of Ether). Complex smart contracts, which within has millions of computations, would require a significant amount of Gwei. Volatility regarding prices and the number of transactions create cost instability for those operating in the network.

3.4 Regulatory Obstacles

Alongside a broad spectrum of technological obstacles, there are regulatory hurdles as well. Unfortunately, the Blockchain ecosystem has been associated with illegal activities. Due to Bitcoin's pseudonymity, criminals have found the nascent technology to be a well-suited medium for money-laundering, illicit trade or to fund terrorist activity[Mal18]. Bitcoin is also a popular payment method for cybercriminals[Nad18].

Illegal Activity

To tackle criminal activity, clear communication standards are needed between those utilizing bitcoin legally, such as cryptocurrency exchanges or regular institutions, and the government. Complex software that can detect irregularities on the network is also in high demand. There are many bright minds in the crypto ecosystem; utilizing these is important for those trying to regulate the network. If governments or other financial institutions were to offer benefits to those who reveal and identify various threats, you could crowd-source endeavour to monitor the network. There are, however, several "privacy coins". These cryptocurrencies are not traceable and can be sent between two accounts, without ever disclosing the transacting parties publicly. These cryptocurrencies are not possible to track or monitor in any way known to this date - an (unfortunately) excellent medium to conduct illegal activities through.

Regulation

During the Initial Coin Offering (ICO) craze of 2017 $6.2 billion was funded to 1258 projects[ICO]. Unregulated ICO's lure the community with big words and false promises. ICO-prices, fueled by investor greed, saw exponential returns, spreading the word rapidly. As the end of 2017 was closing in, investors were borrowing money and taking out second mortgages to buy Bitcoin, causing devastation when the exponential growth was followed by a complete collapse of the cryptocurrency market.

This market collapse is often compared to the dotcom-crash of 2001. While the cryptocurrency charts indeed resemble the aggressive volatility of 2001, the downfall of cryptocurrencies during 2017 have more in common with the *Great Crash* of 1929, that led to the great depression. The government was at this point in lack of regulatory mechanisms. In the wake of the great crash, the government proceeded to create the U.S. Securities and Exchange Commission (SEC). The SEC advocates full public disclosure and protects shareholders against fraudulent and manipulative practices in the market. A similar approach is needed for the cryptocurrency market as well.

Rather than trying to create an instrument to regulate cryptocurrencies, the community saw the downfall of ICO's in 2017 as a sign. Security Token Offering (STO) is technologically almost the same as an ICO. When conducting an STO, the process has to be compliant with the SEC. Combining the stricter regulation of the SEC with blockchain's programmable equity will be advantageous for investors. Another interesting approach called DAICO, which builds upon principles from a DAO and an ICO, is a mean of self-regulation. It forces demands upon project creators, and only releases capital if strict requirements are met[But18].

If blockchain and cryptocurrencies are ever to succeed commercially, strict regulations need to be in place. Regulations that protect investors and force demands upon projects.

3.5 Summary

Blockchain is a complex innovation. During this summary, the most important technological aspects of the blockchain will be presented.

- A blockchain consists of blocks that are chained together utilizing hashing algorithms. It is a continuously updated list of information about the state of the network.
- Validators in the network (miners, forgers or validators) are governing the system through the use of a consensus algorithm.
- A consensus algorithm defines how validators communicate, and ultimately how the network behaves. It is the core structure that aids nodes in the system to reach an agreement on what is the ultimate truth.
- Proof of Work is the oldest, most thoroughly tested consensus mechanism. Other variants include Proof of Stake and Proof of Authority.
- All consensus algorithms have strengths and weaknesses and should be implemented according to the systems' purpose.
- Smart Contracts are pieces of code that reside on the blockchain. These can be interlinked to create complex code structures.

Chapter 4 Related Work

This project is not the first attempt at solving issues regarding counterfeit products. In this section, other projects will be briefly outlined. Although the following projects are blockchain based, they all slightly differ from this contribution.

4.1 VeChain - MyStory

VeChain, founded in 2015, aims to leverage blockchain technology to solve the problem of counterfeit products through enhanced traceability across supply chains and logistics networks. The product range is broad, from wines, agriculture, automobiles to pharmaceuticals [Fou18].

Their blockchain is called "VeChain Thor", which will serve as the backbone for their services. They utilize a consensus mechanism that can be categorized as Proof of Authority, a protocol that is neither fully decentralized nor centralized. This is a model where block producers are not anonymous, but rather 101 known validators. These validators will consist of a broad range of reputable companies. VeChain's consensus mechanism will greatly enhance scalability, although somewhat compromising decentralization.

Their MyStory project aims to employ VeChain Thor as a secure backend, with physical components applied to, e.g. wine bottles. The wine bottles will be tracked throughout their life cycle, and provide end customers with in-depth information regarding key product characteristics such as quality, authenticity, origin, ingredients, water and energy consumption and more.

Unlike MyStory, the PoC created during this thesis can leverage several different blockchains and storage facilities.

4.2 Everledger

Everledger, a London startup intending to combat fraud in luxury markets, such as art, diamonds, valuable minerals and coloured gemstones. Everledger claim they can simplify often complex and fragmented diamond supply chains. Diamonds do not need any additional tags, as 40 distinct data points can identify the stones [Caf15].

Everledger utilizes a hybrid solution that incorporates both private and public blockchains. The public system is based on the Bitcoin blockchain, to ensure immutability of the transaction history.

Unlike their research, this thesis attacks a broader set of products. While luxury markets seem like a natural place to start, this thesis advocates for the importance of securing commodity products as well.

4.3 OriginTrail

OriginTrail, founded in 2013, aims to bring transparency to complex international supply chains [RLD+17]. Their first mission was tackling interoperability challenges faced by the Slovenian food industry. During the year of 2016, the two founders recognized the potential of blockchains. The technology could successfully answer their question of "How do you make sure that data does not get tampered with?". Ensuring transparency and immutable data will enable actors in the network and end users to authenticate products.

Since 2016, OriginTrail has focused on building an open-source protocol on a decentralized network, to bring transparency into international supply chains. This project intends to harness the capability of several blockchain platforms. By diversifying between several blockchains, they can provide high levels of security and availability.

Leveraging several blockchains is done by creating a complex solution stack, with four layers. Application layer, ODN Data layer, ODN Network layer and finally the Blockchain layer. With this stack, the decentralized applications built on top can be more flexible and easier to create. The application layer host management tools for supply chains. The ODN Data layer is a decentralized graph database that connects data between actors in the supply chain. The bottom of the stack leverages core aspects of the blockchain, such as the immutability trait. Data sets from the above layers are immutably fingerprinted on the blockchain using cryptographic hashes.

Unlike the Origin Trail project, this thesis does not focus on creating a solution stack but rather connecting directly from a mobile device, to the underlying blockchain.

4.4 Blockchain Limitations

A significant part of this thesis considers current and future issues that blockchains must overcome. Research question one addresses this issue. There is a lack of related work regarding blockchain limitations. Comparable projects often have limitations as a secondary topic, or an in-depth look at one obstacle, such as the research regarding the lightning network[PD16], or Towards Automation of Smart Contracts[FN16].

However, there are some broader studies into the blockchain limitations, such as M.Swan [Swa15] or D. Drescher [Dre17]. Both books have dedicated chapters to the limitations of current blockchains.

Unlike the papers mentioned in the first section, this research is broader and less in-depth. Contrary to said books, this project combines reading material with a proof of concept - experiencing the limitations from both a theoretical and practical perspective.

Part II

Application Structure, Design and Implementation

Chapter 5 Methodology

To accomplish the envisaged system, the project is divided into four steps. The first step is analyzing and understanding the counterfeit product market and how blockchain technology aspects can aid mitigate the phenomenon. Steps two through four is designing, implementing and validating the proof of concept. The methodology employed is design science research.

5.1 Literature Review - Acquiring Knowledge

As defined in section 1.5, the rundown of the blockchain space is of high importance. A central part of this thesis has been to find and examine information, both in terms of mentioned blockchain technology and product counterfeit. Conducting a thorough literature study creates a satisfactory backdrop in order to investigate further how a decentralized application can be constructed.

Decentralized applications are still immature. During the design and implementation phase of this project, new terminology, definitions and acronyms are encountered. To fully understand the different pieces of technology, the rigor cycle is utilized - see Figure 5.1. Re-iterating back to the reading material to expand the knowledge base is crucial for the final product.

Where Reading Material Was Located

Google Scholar, Google Patents, NTNUopen, IEEE Xplore and Researchgate have been used as search engines for adequate material. Information is also gathered from books supplied by the responsible professor.

Product Counterfeit

Acquiring information regarding the counterfeit goods market is mainly done through books and papers written by actors tightly connected to the topic, such as OECD, EUIPO or Peggy Chaudry. The first two entities have published a jointly conducted report which methodically examines the market of counterfeit products, including patterns, seizure data, provenance economies and impacted industries, communities and nations. P. Chaudry has released several books on the subject and is considered an expert in the field of intellectual property.

Alongside materials published by the mentioned entities, news and other articles provide insight into events regarding counterfeit products. These articles include seizure sizes, typical products and the explanation for the occurrence.

Blockchain Technology

Gathering valuable and accurate information about innovations is challenging. While information regarding blockchains is abundant, sifting through it and removing impurities is where it becomes problematical. A large part of the information gathered is through white papers, forums and articles. The white papers include a great deal of marketing material, while the yellow papers illustrate the technical aspects thoroughly. Unfortunately, yellow papers are somewhat rare. Forums and articles can be cross-referenced to interpret what information is valuable. The lack of peer-reviews and fact-checking reduce the quality of the information gathered.

5.2 Proof of Concept - Applying Knowledge

The importance of building a proof of concept increases as the quality of reading material decreases. As mentioned in the last section, gathering information of high quality regarding blockchains is somewhat tricky. However, the open source code is precious, considering it enables developers to look at existing artifacts in the space.

The creation of a proof of concept is conducted to better comprehend what the blockchain can do, and what it can not. By creating a decentralized application, direct communication between physical components and the blockchain is made possible. The proof of concept exposes blockchain strengths and weaknesses, and create tangible evidence. Concrete results from an implementation combined with knowledge gathered from the background analysis are more robust than solely relying on reading material.

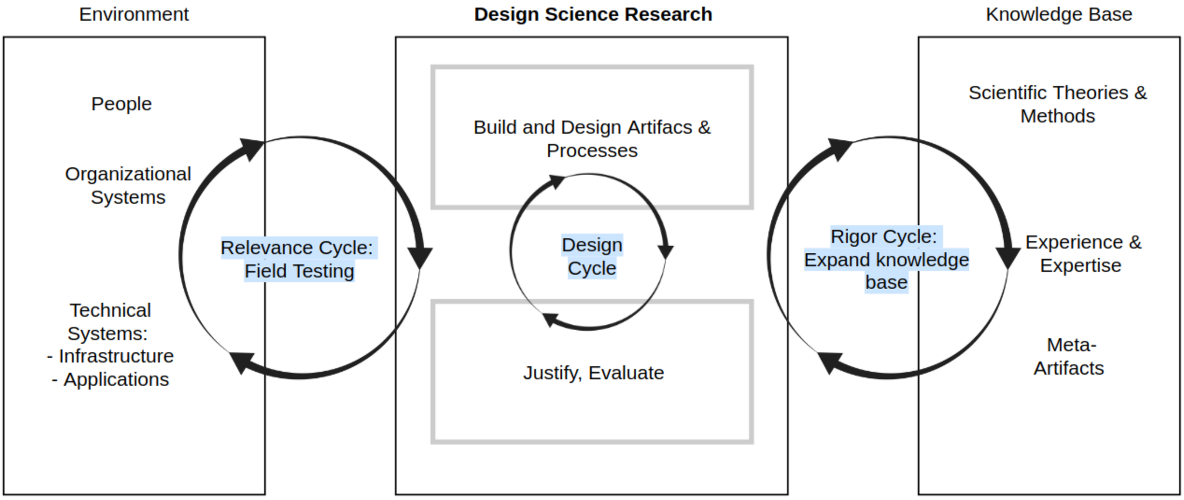


Figure 5.1: The Design Science Research Cycles

Design Phase

In this phase, converting newly acquired knowledge into an actual product was the main focus.

The system was designed to enable intermediaries and end users to interact with immutable storage easily, and by extension, let them trust the data residing in the system. It consists of three main components; an interface, a blockchain and off-chain storage. The system architecture was constructed in this fashion to overcome storage issues addressed by RQ1.2. Off-chain storage is introduced to enable cheap storage while utilizing the blockchain where it excels; immutability and transparency. The exact method is described in chapter 6.

Build Phase

To verify the viability of the envisaged system, a proof of concept application was developed. Various technologies were introduced in order to enhance the quality of the artifact produced. React Native, Ethereum and Google Firebase are central - all thoroughly described in chapter 6.

All code is written in visual studio code, a source-code editor created by Microsoft. By integrating extensions in the editor, it provides support for Solidity (.sol) and javascript (.js) files. The source code was produced with an iterative approach. For each iteration, a new function was implemented. During the implementation, some third parties were introduced. Relying on a third-party such as Infura to provide connectivity with the Ethereum network is only for the sake of simplifying the creation of the artifact - as this can be considered as a single point of failure.

Deploying the smart contract onto the network can be done in several different ways. For testing purposes during the build of this PoC, the Ganache framework (subsection 6.2.7) was utilized. For deploying on the actual network, Remix (subsection 6.2.9) was utilized.

QR-codes were supplied with a JavaScript Object Notation (JSON)-object. The object is then encoded onto a QR-code, which can be printed.

Validation Phase

Evaluation is done continuously throughout the design science research. Validating an iteration of the complete system, including links between components, can only be done after one method has been implemented at all parts of the system.

When adding a method, it is first implemented in the back-end. The smart contract is then deployed on a local blockchain, instantiated by Ganache. Communication with the instance is done through a terminal window, running on the same computer. Several tests are completed to validate that the deployed method is functioning correctly. The validation phase would be extremely inconvenient if it were not for Ganache. When setting up a local blockchain instance, the responses from the blockchain are instantaneous, which facilitate a more agile debugging process.

Once the smart contracts on the Ethereum blockchain are evaluated, the client-side code is mapped against its requirements. The code written is then compiled and deployed onto a smart device running Expo. The device renders any errors onto the screen, making the debugging phase streamlined. When the screen shows satisfying results, the design cycle is continued, adding another function to the smart contract.

After finishing the build, the smart contract was deployed onto the Ropsten test network. Migrating the smart contracts to the Ropsten test network is a process that emulates migration onto the actual Ethereum network, without the cost. The proof of concept was then examined to verify the complete implementation and its conformity to the thesis' goal.

Chapter 6
Proof of Concept Application Design

Similarly to centralized applications, there are different approaches to undertake when designing the software. The structural differences between dApps are many, including to what level a blockchain system is employed. The arguably most successful decentralized applications are the likes of BitTorrent and Email. They employ peer-to-peer communication without utilizing a centralized database to perform their services successfully. These applications themselves are older than the invention of blockchain. This chapter analyzes different approaches to structure decentralized applications, illuminate how blockchain aspects aid the PoC and outline the design of the application. This chapter also functions as a brief introduction to the application design, while chapter 7 goes further in-depth of the different pieces of code.

6.1 Design Choices

The typical blockchain application consists of the blockchain as a back-end and the front end facing the users. Several applications also include some form of off-chain processes, such as user authorization, or the storage of sizeable data structures that are too large for the blockchain to handle.

6.1.1 The Balancing Act

The race towards the perfect scaling solution is in process. Different techniques and approaches yield different results and look increasingly promising, as mentioned in chapter three. In order to process data amounts analogous with traditional databases, several enhancements are needed. State of the art blockchains are inadequate for most business purposes, especially when trying to solve the use case completely on-chain. When blockchain technology combines with off-chain solutions, it enhances scalability over blockchain-exclusive solutions, and augment security compared to conventional centralized database structures alone. OriginTrail, mentioned in chapter four, is an example of a protocol that utilizes aspects from centralized and decentralized structures. While solutions solely based upon blockchain technology might be viable in the not too distant future, a combination appears like the sound choice as of 2019.

In this thesis, a solution that solely utilizes blockchain technology is named "blockchain solution", whereas a system that combines off and on-chain is named a "blockchain hybrid solution". Knowing when to utilize which solution is just as important as understanding what blockchain to base the system on. There are strengths and weaknesses related to both types of systems. Blockchain solutions' pros and cons are thoroughly examined in chapter three. As for the blockchain hybrid solutions, the obvious strength is the scalability of storage.

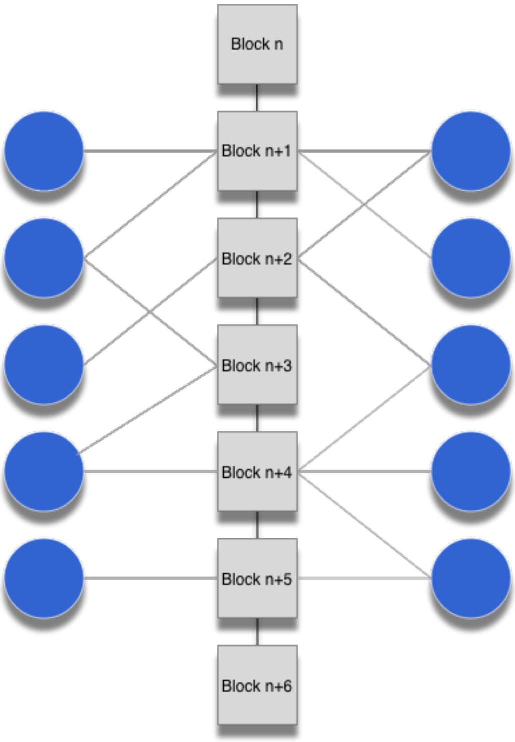


Figure 6.1: Fully decentralized protocol, a **blockchain solution**. Actors (blue circles) interact solely through the blockchain. No off-chain communication.

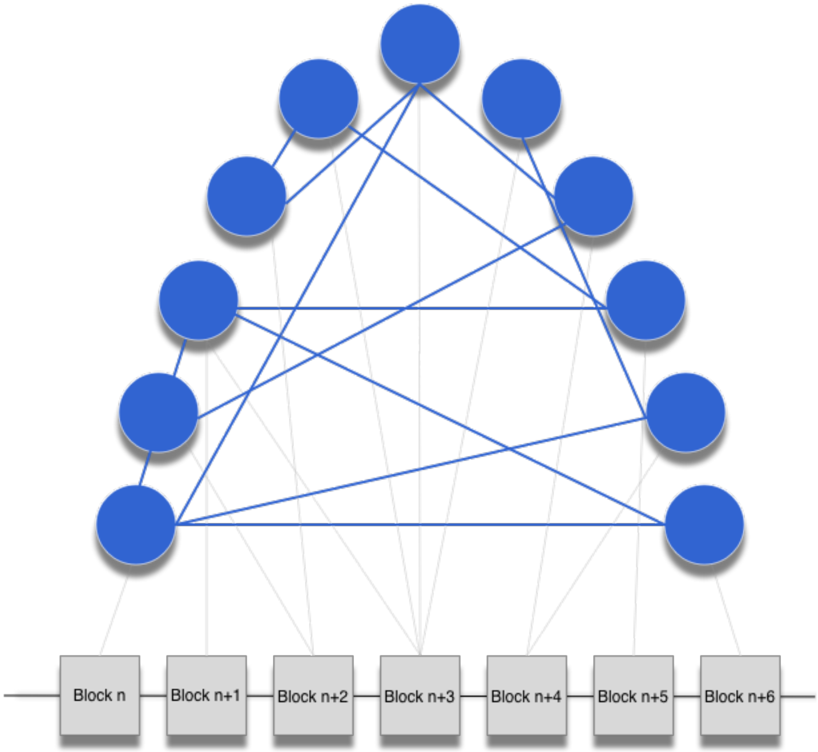


Figure 6.2: **Blockchain hybrid solution**. The protocol allows actors to interact directly with each other. The blockchain can be used for identity authentication, fact-checking and other necessities.

Off-chain storage

When data is stored off-chain, that data is hashed, this hash is stored on-chain. Whenever an actor in the network wishes to withdraw and use this data, it is hashed again and verified against the on-chain hash. This way, the actor can be completely confident that the data is untampered. If the current hash differs from the on-chain hash, the actor is unable to complete its processes. A rigorous back-up system needs to be in place, consisting of a network of databases, or e.g. a solution called InterPlanetary File System (IPFS). The actor is now ensured that **A)** The data is not tampered with, due to the on-chain checksum verification mechanism, and **B)** the real data exist and retain high availability.

The key takeaway from this section is the fact that all messaging, data transfer, data storage and other processes does not *need* to be conducted on-chain. For example, in a Peer to Peer (P2P) market-place, information about the product could be stored off-chain, identification of buyer and seller, as well as payments, are handled on-chain. Utilizing off-chain mechanisms enhances scalability, and although it somewhat compromises security, the sum is a more attractive solution for most purposes.

The PoC created is a balancing act, an exemplification of a "blockchain hybrid solution", employing both on and off-chain storage.

Tailoring the Application for its Needs

This proof of concept in itself does not *need* to store data off-chain since the generated amount is minuscule. When the application scales to the point where thousands of actors are granted write access to the network, and millions of end-users hold read access, there is indeed a need for off-chain storage. While this proof of concept will at no time serve customers, let alone millions of them, it is desirable to compose it in a fashion where it could. Thus information regarding products, such as manufacturing origin, production date, its journey throughout the supply chain and other data can be hashed, saving only the hash on-chain.

6.2 Technologies

Since the PoC employs both off and on-chain storage, several different pieces of technology are needed. Google Firebase and the Ethereum network serves as the back-end. The interface is built utilizing React Native, and several JavaScript dependencies. QR-codes are the physical element of this system. The section introduces the different pieces of technology used.

6.2.1 React Native

React Native is a JavaScript framework based on React, the library constructed by Facebook for the purpose of building user interfaces. React Native is purposed to build mobile platforms, where React is intended for the browser. The platform is known for its "write once use everywhere"-approach, as a significant part of the code written can run on both Android and iOS devices. React Native enables web developers to write mobile applications that feel "native", all the while depending on familiarity to the JavaScript language and libraries.

6.2.2 Expo

Expo [EXP] is a tool that is built to aid developers when creating React Native applications. It provides tools that simplify the whole development process. When utilizing Expo, the developer can quickly inspect the built interface visually on an iOS or Android device.

6.2.3 Google Firebase

Firebase is a system that allows the user to create applications without the need for server-side programming. Firebase supports both iOS and Android, as well as web clients. There are many different services that Firebase introduces, such as authentication, crash reporting, analytics, cloud messaging and storage.

For this PoC, the Firebase service called "Real-time Database" is utilized. This serves as a REST API handling user Hypertext Transfer Protocol (HTTP) requests. "POST", "PUT" and "GET" methods handles information regarding the scanned product.

6.2.4 Ethereum

The core structure of Ethereum is introduced in previous chapters. Earlier mentioned aspects such as security, immutability, trust and transparency are attributes where most public blockchains scores high. Even though Ethereum excels in mentioned attributes, that is not the reasoning behind choosing Ethereum, but rather the community surrounding it.

The large community of developers are helpful, both for aspiring developers, but also in regards to feedback to its creators. The Solidity language is continuously updated, and although unstable as of today - the constant improvements are a positive sign.

6.2.5 Solidity (v0.5.0)

The Solidity language is the most widely adopted programming language for the Ethereum blockchain. It is object oriented and statically-typed, influenced by the likes of JavaScript, C++, Powershell and Python. Its sole purpose is developing smart contracts on the Ethereum network. When Solidity is run, the smart contracts are compiled to bytecode, which is executable by the Ethereum Virtual Machine (EVM).

In 2014, the Ethereum eco-system had several alternatives to create smart contracts, such as Serpent, Mutant and LLL. In 2019 the only two options are LLL and Solidity. For this PoC, the Solidity language is preferable due to the sheer amount of activity, tutorials and developers revolving around it. The current version is 0.5.x - still under development and should "not be considered stable". [SI]

6.2.6 Truffle (v5.0.5)

The Truffle framework is part of the Truffle Suite [TRF] and is a prime example of how a community can help the adoption of technology. Truffle describes itself as a development environment, testing framework and asset pipeline for blockchains using EVM. It aims to simplify dApp development. React Native has its "create-react-native-app"-command [Com] that initiates a project structure with pre-specified folders and files - streamlining the project startup process. The Truffle framework has "truffle init", which initializes a new and empty Ethereum project. A swift setup enables the developer to focus solely on coding, lowering the threshold of smart contract development.

Other essential commands include truffle compile, and truffle migrate. The first compiles the Solidity smart contracts and creates JSON-files. These artifacts are utilized when the contracts are deployed. The second command migrates the contracts onto the blockchain.

Truffle test is another important feature of the framework. The framework enables the developer to write manageable and straightforward test in JavaScript or Typescript. The feature enhances security, as testing become an easier task.

6.2.7 Ganache (v2.0)

Ganache is another significant part of the truffle suite. It is a desktop application that simulates an instance of the Ethereum blockchain, run locally on your machine. When initialized, it creates 10 accounts that the developer can initiate interactions between. Deploying and running smart contracts on a local instance is an approach that helps excel development and testing.

6.2.8 MetaMask

MetaMask considers itself as a bridge that allows users to interact with the decentralized web through a regular web browser. The Ethereum dApps residing on the blockchain are available through MetaMask, without the need to run an Ethereum node (have a copy of the blockchain locally) on your machine. MetaMask provides users with an interface where they can manage identities on the blockchain, and sign transactions.

For this project, MetaMask is used when deploying smart contracts to the Ethereum blockchain. The deployment is itself a transaction. The sender provides the GAS needed, signs the transaction and broadcasts it onto the Ethereum blockchain.

6.2.9 Remix

Remix is a collection of tools which helps developers interact with the Ethereum blockchain. Debugging, testing and deploying contracts are central features. Remix is also considered an Integrated development environment (IDE) since it allows for coding straight in the web browser.

6.2.10 Web3.js (v1.0)

Web3 is the main JavaScript library for interacting with a local or remote Ethereum node. Where many web developers use interaction protocols such as Ajax to interact with a web server, Web3 is used to interact with the Ethereum network.

6.2.11 Infura

Interactions between the client and the Ethereum blockchain are conducted through Web3. To connect to the network, you also need to specify which Ethereum node you want your information from. The safest way to do this is through running an Ethereum node yourself with, e.g. Geth or Parity - "Ethereum clients". Infura provides this service for the developer, omitting the need to set up and run a node.

6.2.12 QR-codes

Quick Response Codes are 2D machine-readable barcodes. The characteristic black and white squares provide unique structures, that can be used as identifiers. QR-codes have several strengths, such as being inexpensive and easy to use.

Finding suitable physical components is not an easy feat. This will be discussed in subsection 9.2.3.

6.3 System Architecture

This section describes how the application is structured and how it is intended to work. The system consists of several different bits, such as the client-side application and the two different storage and computation solutions. Below is an overview of the projects' folder structure.

- src
  - authenticateProduct
    - authenticator.js
  - updateProduct
    - updater.js
    - updateProductOnEthereum.js
    - updateProductOnFirebase.js
  - helperFunctions
    - createProductOnFirebase.js
    - getGeoLocation.js
    - isJson.js
    - updatedCheckSums.js
  - variables
    - ethVariables.js
  - InitFirebase.js
  - Main.js
  - provider.js
- contracts
  - Migrations.sol
  - ProductContract.sol

6.3.1 Client: Mobile application

The front end enables the user to interact with the system. Since most interactions with the system require a camera, it is reasonable to connect through a smart-phone.

All Users

All users of the application hold read-access to the system. In terms of Solidity, the Ethereum programming language, reading information off of the blockchain is done with a "pure" function. This function is completely free - end users are not charged to authenticate products or read any other information.

Intermediaries

Intermediaries that control the flow of products through the supply chain enjoy enhanced access. At every transit point in the supply chain, intermediaries process the products. During this process, the goods are scanned, and information regarding the product is stored. This requires write-access, which all intermediaries hold.

6.3.2 Servers

Ethereum Blockchain

In a nutshell, the Ethereum blockchain is a back end system that, as of 2019, compromises on scalability and responsiveness to acquire a high level of security. Utilizing web3, the service mentioned above, the front end can interact with the Ethereum blockchain through its API, as if it were a regular back-end. Once the client scans a product, it signals the Ethereum blockchain that a product is about to be updated/authenticated, the smart contract fulfills its purpose, and answers the client that updates are in motion.

Off-chain: Firebase Server

The client also interacts with the firebase server. The product information is stored as JSON-entries in a long list of products. As more intermediaries process products, an array is updated. The array stores information about when, where, and who processed the product. The information is hashed at every process point, and the checksum is stored on the Ethereum blockchain.

Intertwined Functionality

The server side is a complex piece of code. Integrating two completely different architectures means dealing with different response times, scalability and security. Concurrence issues are present, and a rigid code structure is important to keep the storages in harmony.

Chapter 7

Decentralized Application Implementation

The implementation of a dApp is complicated, particularly when combining two completely different structures, Firebase Realtime Database and the Ethereum blockchain. This chapter goes in full depth of how the code works, how the two servers operate separately, and how information is gathered, delivered and stored.

The following section introduces how the user interacts with the system and how information flows through it. The last section in this chapter analyzes the communication between the smart contract and the react-native interface.

7.1 React Native Interface - Three Main Processes

The application supports three basic processes. The creation, update and authentication of a product. The front end of the application has two buttons to switch between create/update and authenticate. These functions are completely independent - there is no reason to have them residing on the same application, but for this PoC it is done for convenience. The following sections cover these processes, explain the occurring events, and how they are handled.

7.1.1 Creating and Updating Products

The manufacturer and the intermediaries along the supply chain employ the mentioned processes. When a manufacturer produces a product, a QR-code is attached. Residing in the QR-code is information about the product, such as manufacturer, origin, description and ID. The manufacturer scans the QR-code; information is gathered and delivered to the two servers.

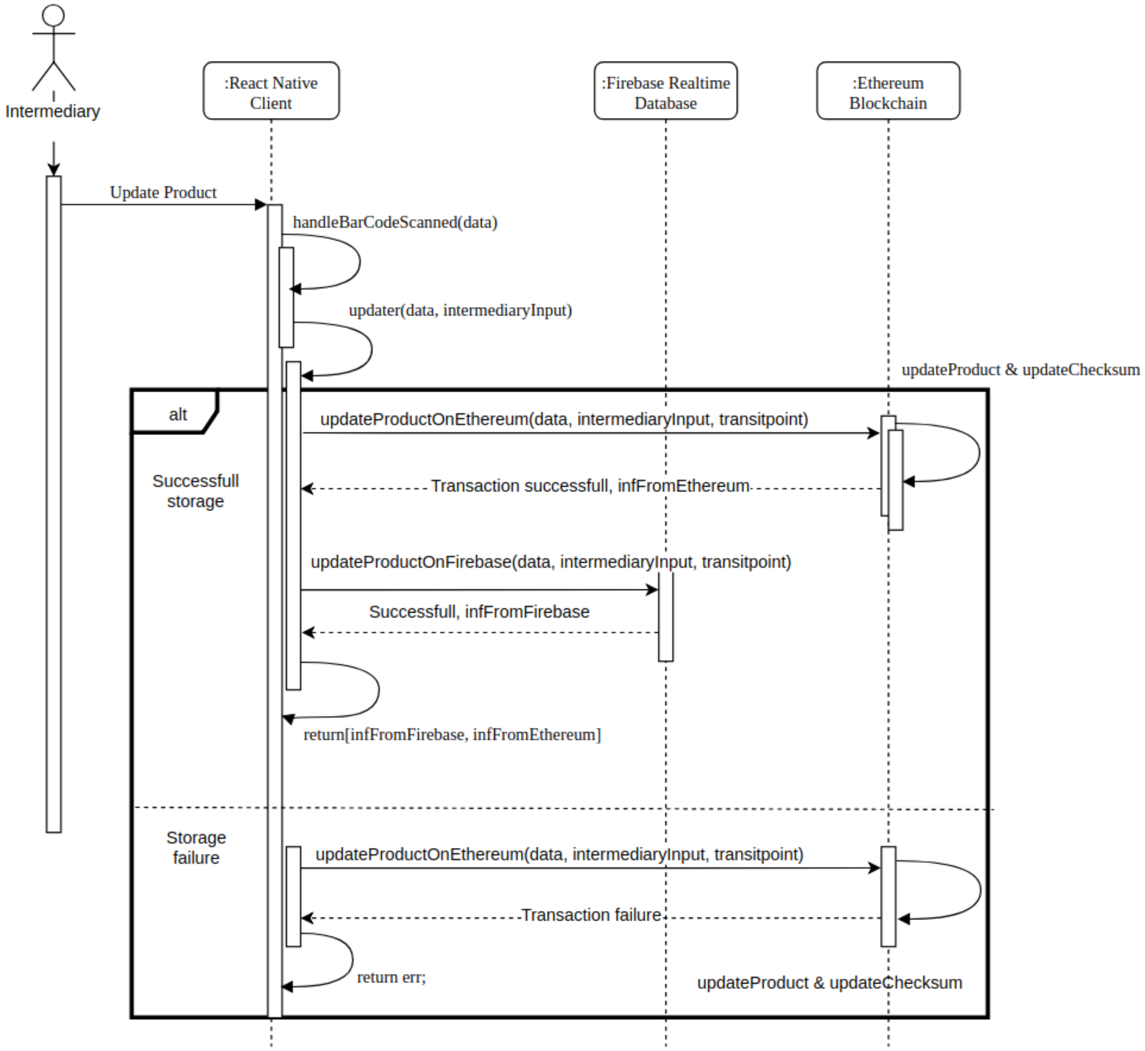

Intermediary
:React Native Client
:Firebase Realtime Database
:Ethereum Blockchain
Update Product
handleBarCodeScanned(data)
updater(data, intermediaryInput)
updateProduct & updateChecksum
alt
Successfull storage
updateProductOnEthereum(data, intermediaryInput, transitpoint)
Transaction successfull, infFromEthereum
updateProductOnFirebase(data, intermediaryInput, transitpoint)
Successfull, infFromFirebase
return[infFromFirebase, infFromEthereum]
Storage failure
updateProductOnEthereum(data, intermediaryInput, transitpoint)
Transaction failure
return err;
updateProduct & updateChecksum

Figure 7.1: Sequence diagram showing an overview of the information flow of updating a product. The flow for creating a product has negligible differences. Below follows a detailed explanation.

handleBarCodeScanned

The function named *handleBarCodeScanned* (7.1) is an operation that is called upon when the camera detect any QR-code it is pointed at. It gathers the information, checks whether the data is JSON or not, and alerts the user about the scanned information. If the scanning process is unsuccessful, the intermediary is alerted and provided information on why it failed. If the scan is successful, the intermediary is alerted with the product ID and asked whether or not it would like to continue with the updating process.

updater

Once the intermediary continues with "OK" the information is handled by a function named *updater*. The function accepts two parameters; scanned data, and the intermediary name - which functions as the intermediary ID in this PoC. The updater parses the data provided from the scanned QR-code, and gather the information regarding the intermediary, the name, geolocation and a timestamp. The updater will call upon two functions; *updateProductOnEthereum* and *updateProductOnFirebase*.

updateProductOnEthereum

This function has three parameters, the parsed data, the transit point including location and timestamp, and the intermediary name. When called upon, the function connects to the Ethereum Blockchain - the Ropsten Test network in this PoC. It creates a transaction and calls upon a method named *updateProduct* residing on the smart contract named "ProductContract".

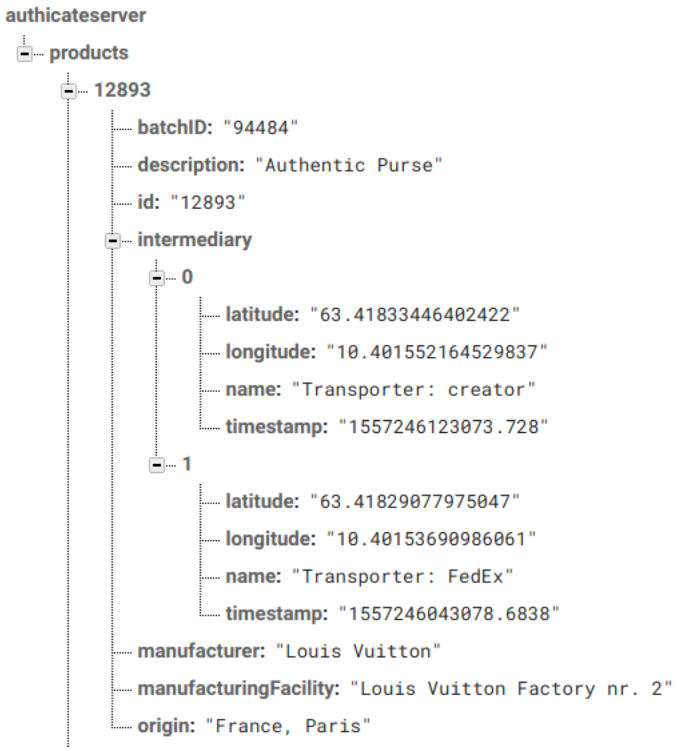
mapping(string => bytes32[]) private products;

This mapping has string keys, which is the product IDs. The key map to a value which is an array with entries of the type "bytes32". Depending on whether or not the product ID already exists in the mapping, *updateProductOnEthereum* will either 1. produce a new entity in the mapping, or 2. add a new element in the array. The "updateProduct" calls upon a second SC function called *updateChecksum* - which creates and returns a hash of the information provided. This hash is of type bytes32 and is added to the array in the mapping. This hash is a checksum, which is used in the product authentication processes.

updateProductOnFirebase

The second function call in the *updater* is a call to the Firebase server. This function accepts the same parameters as *updateProductOnEthereum*, and will either 1. create a new product entry, or 2. add a new element in the "intermediary" array, that exists within a product on firebase. Firebase will store data as JSON in its Realtime Database, as seen in Figure 7.2

Once finished, the *updater* returns information about the product to the interface, so that the intermediary can inspect the information stored.



```
authicateserver
  products
    12893
      batchID: "94484"
      description: "Authentic Purse"
      id: "12893"
      intermediary
        0
          latitude: "63.41833446402422"
          longitude: "10.401552164529837"
          name: "Transporter: creator"
          timestamp: "1557246123073.728"
        1
          latitude: "63.41829077975047"
          longitude: "10.40153690986061"
          name: "Transporter: FedEx"
          timestamp: "1557246043078.6838"
      manufacturer: "Louis Vuitton"
      manufacturingFacility: "Louis Vuitton Factory nr. 2"
      origin: "France, Paris"
```

Figure 7.2: JSON-object structure stored on Firebase

7.1.2 Authenticating Products

The first two features, update and create, are strictly purposed for intermediaries and manufacturers respectively. The third feature is where the end users can authenticate products, whether it is NASA, Microsoft, a vaccination firm or a person with new sneakers.

handleBarCodeScanned

The same function that handles the update/create-process handles the authentication process. An if-sentence checks whether the user interfacing the system has chosen "update" or "authenticate". In a production version of this PoC, separation of these processes is reasonable.

After the initial check, the function calls upon a function named *authenticator*.

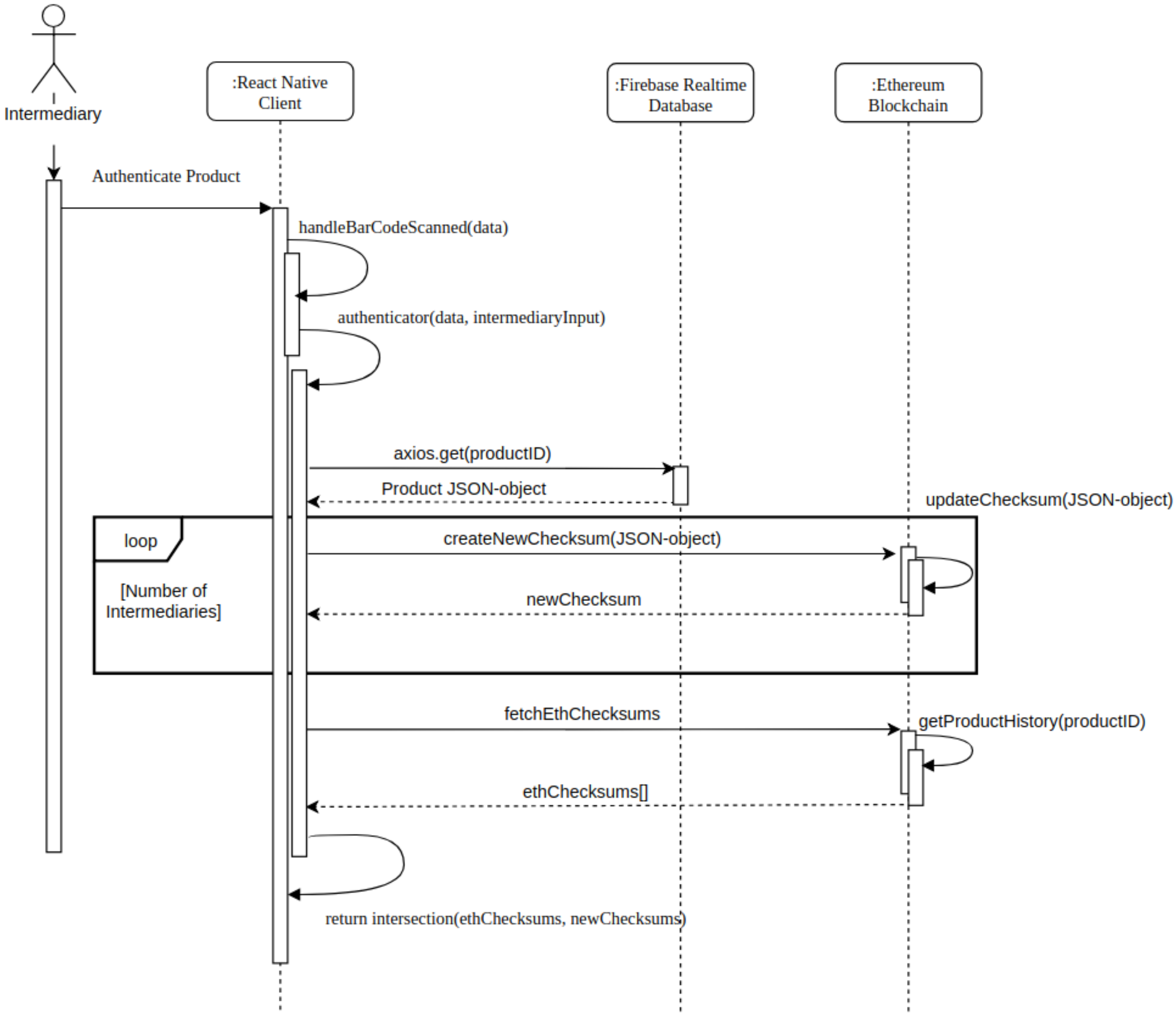

Intermediary
:React Native Client
:Firebase Realtime Database
:Ethereum Blockchain
Authenticate Product
handleBarCodeScanned(data)
authenticator(data, intermediaryInput)
axios.get(productID)
Product JSON-object
updateChecksum(JSON-object)
loop
[Number of Intermediaries]
createNewChecksum(JSON-object)
newChecksum
fetchEthChecksums
getProductHistory(productID)
ethChecksums[]
return intersection(ethChecksums, newChecksums)

Figure 7.3: Basic information flow when authenticating a product.

authenticator

This function accepts one parameter, which is the data scanned by the user. Firebase utilizes the data-parameter fed from the QR-code when a user requires information about the product. Firebase responds with an object, which has an element called "data". If "responseFromFirebase.data" equals null, the product does not exist in the firebase database, and the user is informed.

If firebase locates the product, a variable in the application code stores the responding JSON-object.

In the response, fields such as batchID, description, id, manufacturer, intermediary and origin is shown to the user. These fields are utilized when authenticating the product. "Intermediary" is the most interesting field in the JSON-object, as it contains an array with entries of intermediaries that have processed the product on its way from inception to end user.

createNewChecksum & fetchEthChecksum

When authenticating a product, the *authenticator* calls upon two functions, *createNewChecksum* and *fetchEthChecksum.* The first function creates an array of new checksums created based on the JSON-response in Figure 7.2. The latter function fetches checksums from the "products" mapping, with a call to the smart contract function *getProductHistory* - a function that returns every checksum created for a certain product.

The two arrays of checksums, (1) created by *createNewChecksum* and (2) fetched from Ethereum with *fetchEthChecksum*, are then matched. If no information is altered in the firebase database, every entry in the newly created array (1), should have a matching entry in the reference values array (2). Information regarding the authenticity of the products is then shown to the user.

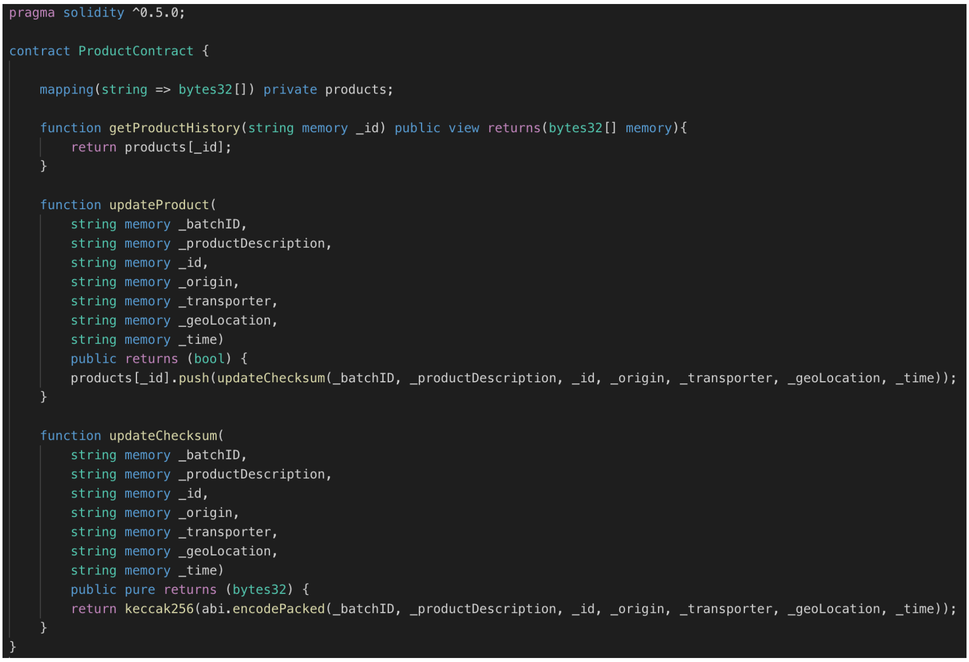
Response To User

Instead of just showing the user **authentic** or **not authentic**, the interface is given information about *how many* matching entries there are. If information about, e.g. who scanned a certain product is altered, the application shows the user that one of the intermediary entries is not deemed authentic. A typical response would be "Out of the 35 transit points this product has been processed at, it is deemed authentic at 34".

The feedback to the user is also given in the form of a visual response - a map, as shown in chapter 8.

7.2 Ethereum Blockchain - *productContract*

For this project, one smart contract called *productContract* handles all blockchain operations. It is relatively small in size, and at this stage, there is no need to divide the contract to achieve easily comprehensible logic.



```
pragma solidity ^0.5.0;

contract ProductContract {

    mapping(string => bytes32[]) private products;

    function getProductHistory(string memory _id) public view returns(bytes32[] memory){
        return products[_id];
    }

    function updateProduct(
        string memory _batchID,
        string memory _productDescription,
        string memory _id,
        string memory _origin,
        string memory _transporter,
        string memory _geoLocation,
        string memory _time)
        public returns (bool) {
        products[_id].push(updateChecksum(_batchID, _productDescription, _id, _origin, _transporter, _geoLocation, _time));
    }

    function updateChecksum(
        string memory _batchID,
        string memory _productDescription,
        string memory _id,
        string memory _origin,
        string memory _transporter,
        string memory _geoLocation,
        string memory _time)
        public pure returns (bytes32) {
        return keccak256(abi.encodePacked(_batchID, _productDescription, _id, _origin, _transporter, _geoLocation, _time));
    }
}
```

Figure 7.4: The smart contract residing on the Ethereum blockchain. It contains three functions and one mapping

7.2.1 Deploying the Smart Contract

There are various options to consider when deploying a smart contract, including deploying it through Remix or Geth, mentioned in subsection 6.2.9 and subsection 6.2.11, respectively.

For this PoC, Remix is the chosen approach. Utilizing Remix to deploy smart contracts enables the developer to migrate the contract onto the blockchain easily.

After deployment, if successful, the smart contract resides on an Ethereum address - a "contract address". All communication made with the smart contract utilizes the contract address as a gateway.

7.2.2 Communication

The Web3 javascript library explained in subsection 6.2.10, provides a communication link between the front end and the smart contract.

The smart contract address is saved as a variable in the "ethVariable"-folder[section 6.3], along with other important variables such as the contract ABI. The Application

Binary Interface (ABI) is used to encode/decode solidity code to communicate with the EVM.

"Read"-transactions

Two of the previous three functions are "pure" functions. Calls to these functions from the react native application do not write to the blockchain and hence do not need to include GAS when sent. They are chargeless, an important feature. It cripples the system if those authenticating their products have to pay to do so.

"Write"-transactions

When writing to the blockchain, the caller of the function creates a transaction object. First, the caller needs to know the current amount of transactions that the address has ever sent. The transaction count is set as a "nonce" in the transaction. Second, the caller sets the "gasLimit" and "gasPrice", which is the max amount of gas used, and the price for that gas respectively. When interacting with a smart contract, the caller needs to know the contract address - which forms the "to" parameter field. Lastly, the caller must know what method he wants to call, and also what parameters the method accepts.

When constructing the transaction object, the caller needs to sign it with his private key. The transaction is then serialized and converted to "raw" format - a hex-representation of the data residing in the transaction. Then finally, the transaction is sent with a call to an Ethereum function named "sendSignedTransaction".

7.2.3 Functions

getProductHistory: accepts one parameter - the product ID. It returns the complete history of the product with matching ID. The function is utilized when authenticating products.

updateProduct: accepts seven parameters to update a product, e.g. who is updating it, their location and the current time. This function calls upon *updateChecksum*, and pushes the result into the **products** mapping.

updateChecksum: accepts the same seven parameters as *updateProduct*. The react native front end also calls upon this function when authenticating a product. updateChecksum is a "pure" function.

Part III

Results, Discussion and Conclusion

Chapter 8 Results

This chapter outlines the results of this study. The contributions include the essential information gathered from the background analysis and the decentralized application.

8.1 Background Analysis

The background analysis in Part I, provides a thorough investigation of product counterfeit, as well as a study of blockchain technology.

8.1.1 Product Counterfeit

- Product counterfeit is an ancient phenomenon and exists wherever trademarks and other IP exists
- Mostly weak legal frameworks employed
- Extreme amounts of goods flowing through transit points, which lessen possibilities of seizing forged goods

As mentioned in chapter 2, the complexity of supply chains created by enhanced pace and span, information unavailability and products' inability to be easily authenticated identifies as some of the most pressing issues. These problems exist based on several factors:

- There is currently a reluctance to share data with other actors in supply chain networks, defined as silo mentality
- Mentioned reluctance is fueled by current systems' inability to provide the possibility of "open data", allowing entities to interact in a secure, efficient matter

- Authenticating products is often done visually, looking for traits that coincide with authentic products - current authentication processes are out of date
- End users do not have the possibility of auditing information regarding their products' life

8.1.2 Blockchain Technology

This section relies on information found in chapter 3, as well as new information acquired during the implementation of the application.

Scalability regarding storage, throughput and transaction speeds is arguably the most pressing issues regarding blockchains. There are different techniques to enhance scalability, including layer-2 solutions or altering the core protocol. Layer-2 solutions such as the Lightning and Raiden networks boost responsiveness and throughput, but re-introduces the possibility of third parties, creating a hub-and-spoke topography. Alterations at the protocol level, such as modified block size or shorter inter-arrival time could greatly enhance responsiveness and throughput but would centralize the mining operation. It would also create a growth problem - as the total blockchain size could increase exponentially. Sharding is another proposed technique that allows miners to verify a subset of transactions in the network, which would significantly enhance transaction speeds.

The four consensus mechanisms identified in chapter three are some of the most promising utilized in blockchains. PoS, PoA and dPoS all have clear advantages over PoW regarding scalability. PoW is however thoroughly tested over the last decade, and by many regarded as the most secure version.

The "critical code" residing in smart contracts calls for extremely accurate coding. Different approaches to achieve high levels of security include automated generation of smart contracts, utilizing building blocks and third-party auditing.

8.2 Decentralized Application

The decentralized application consists of the hybrid back-end structure, and the interface utilized to interact with the system. The created PoC mainly answers RQ2, but it also exposes the most pressing issues identified in RQ1.

8.2.1 Back-end System

Most applications do not leverage the advantage the blockchain offers. Most decentralized applications do not leverage the advantages of cheap storage and privacy, which traditional centralized databases provide in abundance. The contribution is thus implementing a dApp that takes advantage of both structures. The result is a proof of concept that demonstrates such an intertwined back-end. The implementation exists on the Ethereum test-network and utilizes the Firebase Realtime Database as the centralized service. Both the Ethereum smart contract and the centralized database is currently accessible, with read and write access provided for those who want to analyze the system.

Intertwined Structure

The PoC utilizing a blockchain hybrid solution provides critical services for those wanting to employ it. Provided by the "immutability"-trait of the Ethereum blockchain, to the PoC:

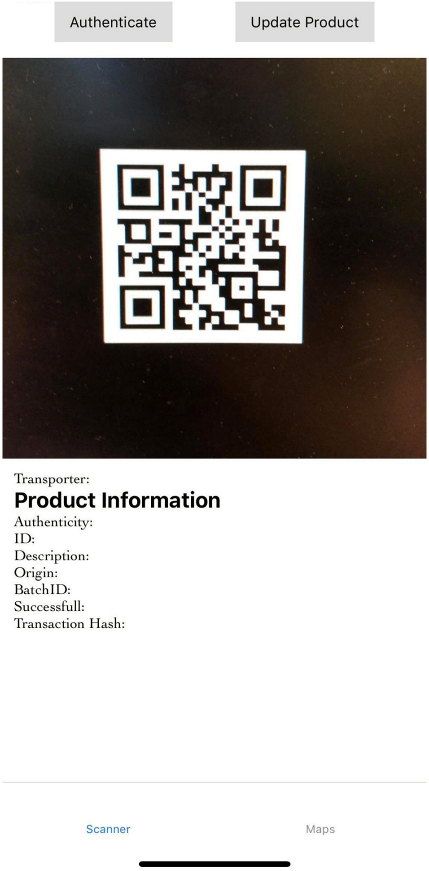
- Manufacturers and intermediaries can leverage the smart contract to store fingerprints of digitized information, with a guarantee against fingerprint alterations
- End users can audit a digitized trail of information regarding their products, and fully trust the authentication process and its results

Provided by the intertwined database structure to the PoC:

- Manufacturers and intermediaries can store data, with a guarantee that that the data cannot be altered without signalling actors in the network
- By leveraging cheap, centralized storage, manufacturers and intermediaries can store rapidly growing amounts of data in the system, without being exposed to an exponential growth of cost

Front-end Interface

As earlier mentioned, this interface includes both the authenticate and update/create-functionality, for convenience sake. All information regarding the scanned product is shown to the actor interacting with the system, as displayed in Figure 8.1.



Transporter:

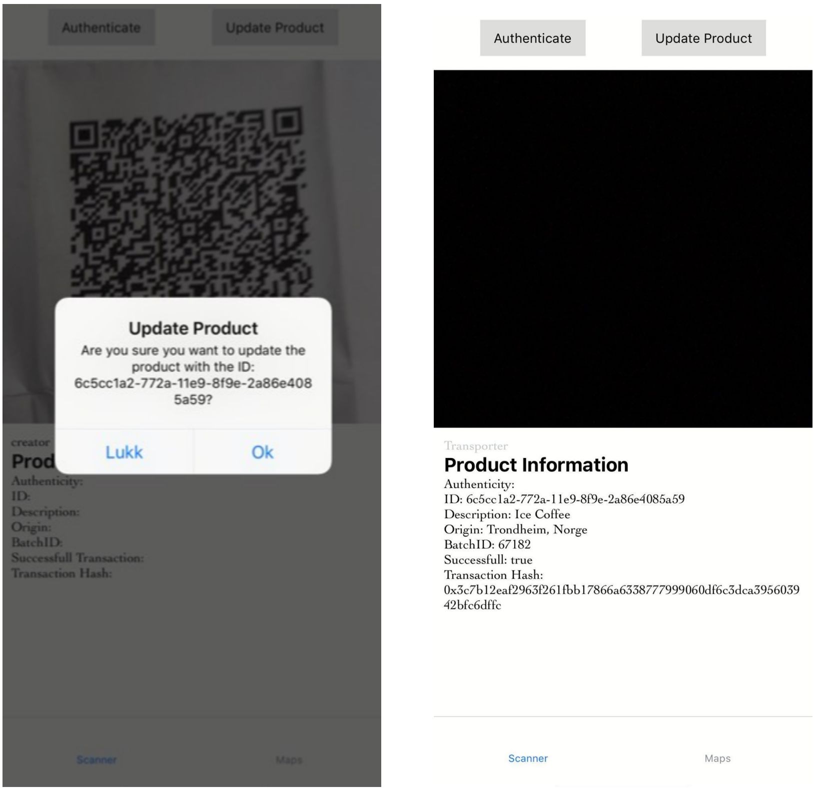
Product Information

Authenticity:
ID:
Description:
Origin:
BatchID:
Successfull:
Transaction Hash:

Figure 8.1: Simple interface.

Update

When updating or creating a product, the "update"-button is pressed, and the mode is changed. The actor updating the product provides his ID - the name of the current transporter.



Update Product

Are you sure you want to update the product with the ID: 6c5cc1a2-772a-11e9-8f9e-2a86e408 5a59?

creator

Prod

Authenticity:
ID:
Description:
Origin:
BatchID:
Successfull Transaction:
Transaction Hash:

Transporter

Product Information

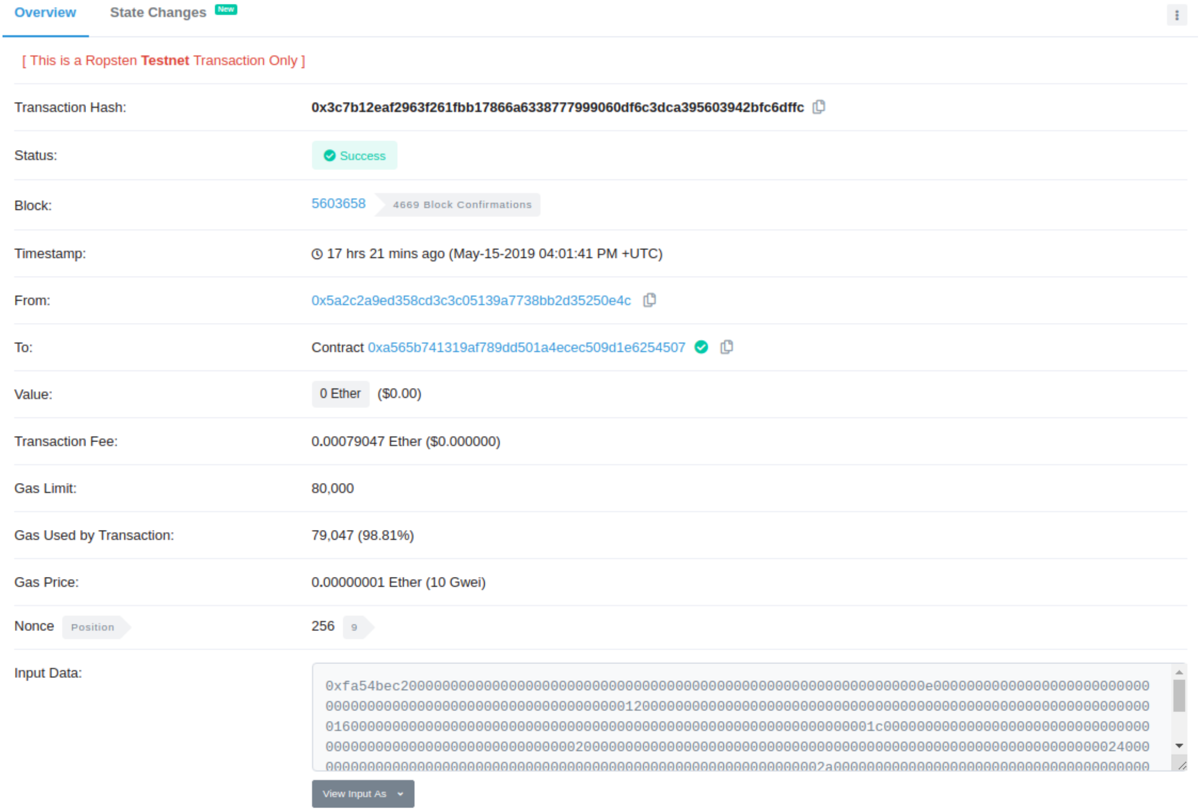
Authenticity:
ID: 6c5cc1a2-772a-11e9-8f9e-2a86e4085a59
Description: Ice Coffee
Origin: Trondheim, Norge
BatchID: 67182
Successfull: true
Transaction Hash:
0x3c7b12eaf2963f261fbb17866a6338777999060df6c3dca3956039
42bfc6dffc

(a) When in "update"-mode, the intermediary is asked if it wants to update the scanned product

(b) Information seen after successfull update

Figure 8.2: The update process as seen by an intermediary.

After pressing "OK", the application updates the product with the given ID. Then, when Ethereum and Firebase complete the update, the intermediary gets a confirmation in the form of product and transaction information. The "Successful"-variable shows whether or not a transaction is considered successful. The transaction hash is inspectable on the Ethereum network, as seen in Figure 8.3.



[This is a Ropsten **Testnet** Transaction Only]

Transaction Hash: **0x3c7b12eaf2963f261fbb17866a6338777999060df6c3dca395603942bfc6dffc**

Status: Success

Block: 5603658 4669 Block Confirmations

Timestamp: 17 hrs 21 mins ago (May-15-2019 04:01:41 PM +UTC)

From: 0x5a2c2a9ed358cd3c3c05139a7738bb2d35250e4c

To: Contract 0xa565b741319af789dd501a4ecec509d1e6254507

Value: 0 Ether ($0.00)

Transaction Fee: 0.00079047 Ether ($0.000000)

Gas Limit: 80,000

Gas Used by Transaction: 79,047 (98.81%)

Gas Price: 0.00000001 Ether (10 Gwei)

Nonce Position 256 9

Input Data:

```
0xfa54bec200000000000000000000000000000000000000000000000000000000000000e00000000000000000000000000
000000000000000000000000000000000000120000000000000000000000000000000000000000000000000000000000000
016000000000000000000000000000000000000000000000000000000000000001c00000000000000000000000000000000
0000000000000000000000000000002000000000000000000000000000000000000000000000000000000000000000024000
000000000000000000000000000000000000000000000000000000000002a00000000000000000000000000000000000000
```

Figure 8.3: Transaction inspection.

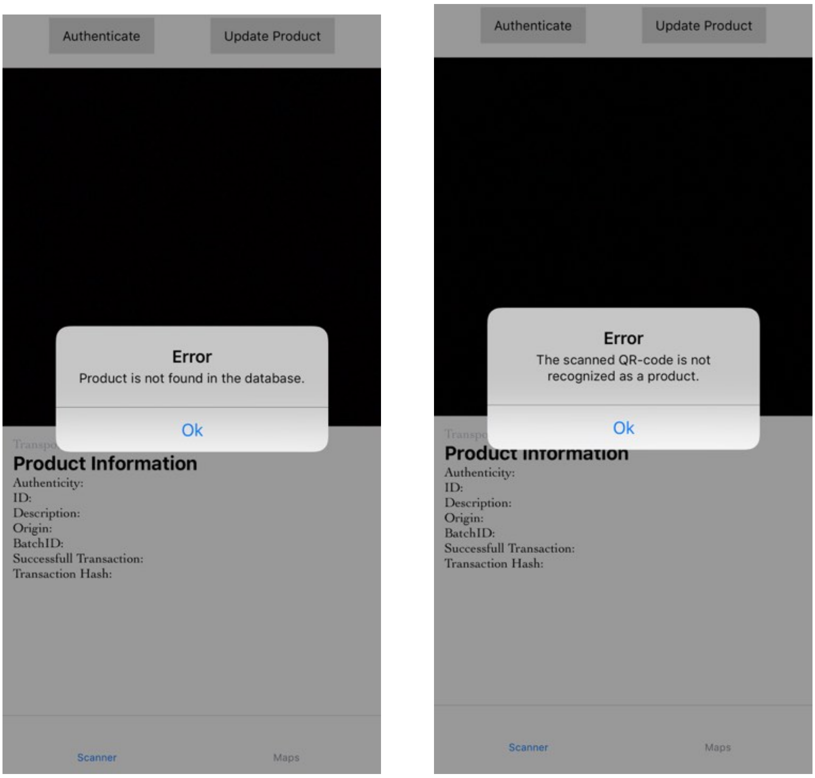
As the figure shows, this is only a Ropsten Testnet transaction. The transaction is successful and located in block 5603658 of the Ropsten Testnet. The transaction is sent from the address of the application to the smart contract address. The value sent is zero, but the transaction is subject to a fee. This fee is 0.00079047 Ether (~0.17 USD at the transaction event), which is a substantial amount for one update. This might however not be representative of the typical cost, as the "Gas Limit" is set quite high for this transaction. "Input Data" is the actual data sent to the smart contract. The data is open and transparent - anyone can look at the transaction and extract information.

After the update concludes, the intermediary can continue the processing of other products. The same procedure is repeated until all products are scanned.

Authenticate

When the product has reached its destination, it is scannable for those wanting to establish the product as genuine.

QR-codes are used for a wide variety of applications. For this PoC, there are some features implemented to ensure that the database only handles QR-codes related to the application. Figure 8.4 demonstrates two error messages displayed to the user while trying to authenticate a specific product.



Error
Product is not found in the database.

Product Information
Authenticity:
ID:
Description:
Origin:
BatchID:
Successfull Transaction:
Transaction Hash:

Error
The scanned QR-code is not recognized as a product.

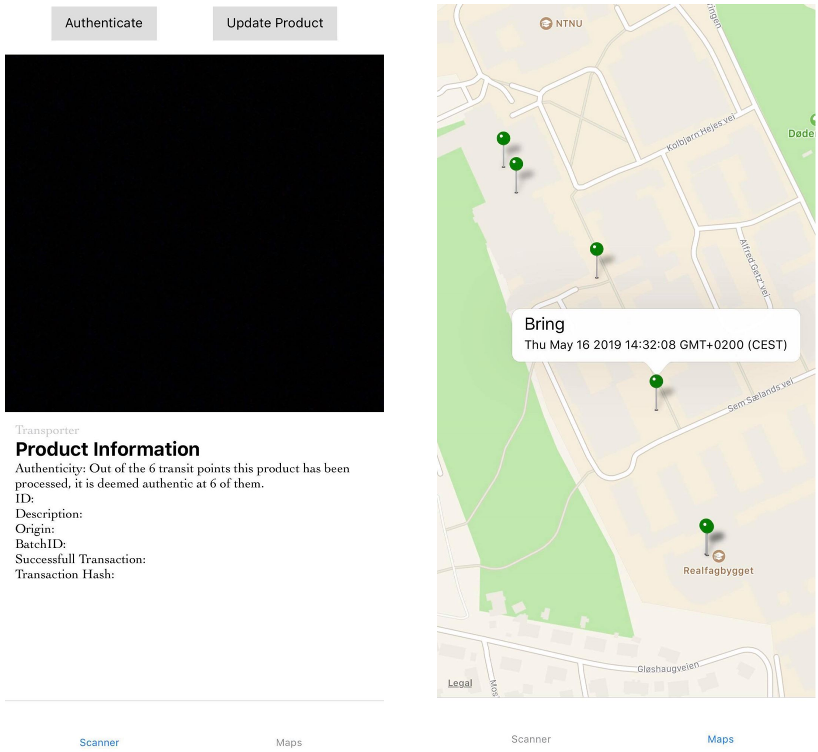
Product Information
Authenticity:
ID:
Description:
Origin:
BatchID:
Successfull Transaction:
Transaction Hash:

(a) If the QR-code is not recognized as a product, error is shown in both modes.

(b) The actor is informed if the product does not exist.

Figure 8.4: Errors during authentication

If the QR-code is associated with an actual product, the user is given visual feedback in the form of text and a map. The text is found under "Authenticity" in the "Scanner"-interface, as shown in Figure 8.5



Transporter

Product Information

Authenticity: Out of the 6 transit points this product has been processed, it is deemed authentic at 6 of them.
ID:
Description:
Origin:
BatchID:
Successfull Transaction:
Transaction Hash:

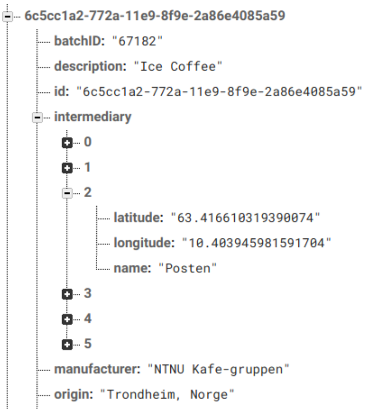
Bring
Thu May 16 2019 14:32:08 GMT+0200 (CEST)

(a) Number of transit points are verified (b) Which transit points are verified

Figure 8.5: The authentication process seen by the end user.

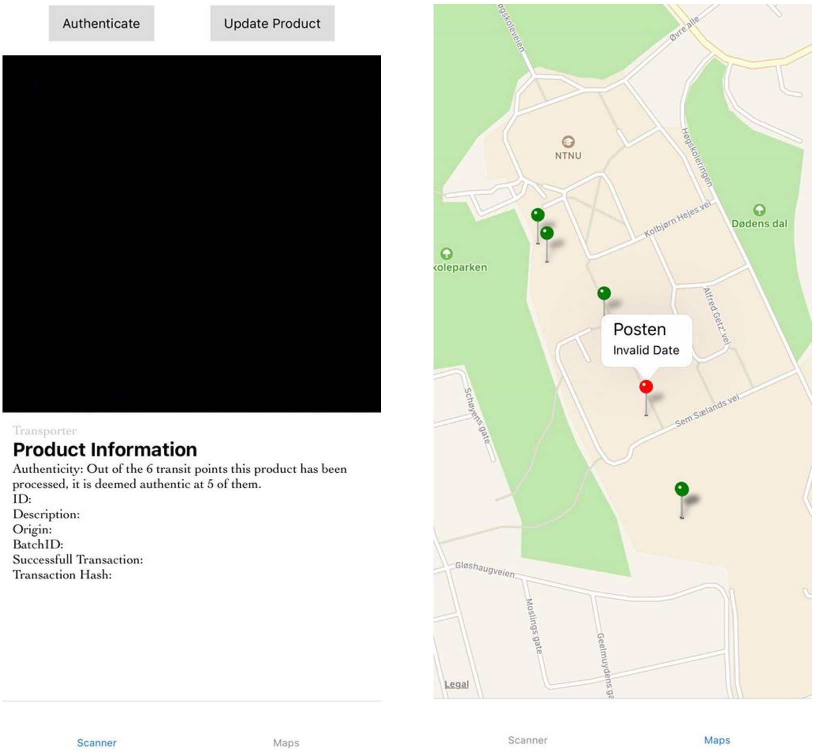
The text feedback provides the user with a fraction that highlights the number of verified transit points, while the map shows precisely which transit points are verified. The user can inspect each green pin, to gather information about when, where, and who processed the product. This can be done for all pins, from point of sale to the manufacturer, allowing the user to audit the chain of transit points.

Green pins equal verified transit points. If the information experience alterations, the pin turns red. As shown in Figure 8.6 the state is altered. The transporter is changed from "Bring" to "Posten", and the timestamp has been deleted.



```
6c5cc1a2-772a-11e9-8f9e-2a86e4085a59
    batchID: "67182"
    description: "Ice Coffee"
    id: "6c5cc1a2-772a-11e9-8f9e-2a86e4085a59"
    intermediary
        0
        1
        2
            latitude: "63.416610319390074"
            longitude: "10.403945981591704"
            name: "Posten"
        3
        4
        5
    manufacturer: "NTNU Kafe-gruppen"
    origin: "Trondheim, Norge"
```

Figure 8.6: Altering the state of the database.



Product Information

Authenticity: Out of the 6 transit points this product has been processed, it is deemed authentic at 5 of them.
ID:
Description:
Origin:
BatchID:
Successfull Transaction:
Transaction Hash:

(a) One of six transit points is tampered with.

(b) The **red** pin shows "Posten", and no date.

Figure 8.7: Responses after altering the database.

When the authentication process is re-run after altering, the result would be as shown in Figure 8.7.

Chapter 9 Discussion

This chapter includes an evaluation and discussion of this thesis' results. Various subjects have been central in shaping the proof of concept. These topics will be discussed.

A recurring theme throughout this thesis is the fact that state of the art blockchains need alterations to facilitate widespread adoption from businesses. The results from the background study show that the obstacles are addressed by central actors and the community around them. However, solving the current obstacles is challenging, as almost every alteration compromises on the attributes that make blockchains attractive.

9.1 Technical Modifications

Research question one asks what technical modifications are needed to make current blockchain solutions viable for businesses. Smart contract security, scalability and consensus algorithms are three central aspects that need advancements. Privacy is important for certain large entities. Two projects are identified, either on-chain (Zether Smart Contracts) or outsourcing privacy to another platform, such as Enigma. While interesting projects, they are yet to be implemented at a large scale. Price stability is another key issue. Large financial institutions, which can run through billions of computations every day, cannot produce budgets on the current network structure since the price of computation varies. No business owner is willing to operate applications or a business on a blockchain, or anywhere, at an unpredictable and unstable cost. A great deal of the costs originates from the need to store large amounts of data. If actors utilize off-chain storage, exposure to price instability is reduced.

RQ1.1

Improving smart contract security is more about providing strict guidelines while developing than just technological advancements. Auditing, automation of smart contract creation and utilizing building blocks are three identified measures to enhance security levels.

RQ1.2

The scalability issue can be split into three categories: storage, throughput and latency. At this current time, saving large amounts of data at a reasonable price level is made possible by introducing off-chain storage. If all data is to be saved on the blockchain, there is a need for significant updates at the core protocol level. As discussed, several proposals have been made to scale throughput, but most modifications make crippling compromises such as accelerating the growth of blockchain size or centralizing mining operation. Future endeavours are needed by developers to scale throughput at the protocol level. Where throughput and storage seem like obstacles that can be overcome, latency appears as a bottleneck. Shorter block times is one way to enhance latency, but make the same crippling compromises mentioned above. Layer-2 solutions enhance throughput and latency but re-introduce intermediaries. Escalating latency to the point of centralized databases is improbable.

RQ1.3

This objective is addressed in subsection 3.2.2. Currently, PoW is utilized by the two largest blockchains in the space. The mechanism is thoroughly tested, which has revealed several issues. As the pool distribution figure in chapter 3 show, mining is now centralized to a few actors that control the network. The increasing adoption of cryptocurrencies and the resulting growth in transactions has made it evident that current proof-of-work consensus mechanisms are unable to adapt to mainstream use. PoS, PoA and dPoS all have advantages over PoW when it comes to scalability. They are, however, not as thoroughly tested. Ultimately, the choice of blockchain, and by extension, the consensus mechanism, comes down to what purpose the application has.

9.2 The Decentralized Application - Strengths and Weaknesses

This section focuses on the proof of concept and mainly discusses RQ2. However, it also exposes some issues discussed in the section above.

9.2.1 Open Data

Enhancing the bandwidth for information flow between entities is important. Currently, interactions between software controlled by different entities are often done with some form of Application Programming Interface (API). API's are highly configurable and mostly controlled by one entity. Rather than trying to make separate software interact fluently, building software that enables entities to store a **shared state** seamlessly, is desirable. This system enables that interoperability between separate entities, without the need to translate information residing on one database so that it fits specifications of another. The ease of sharing data can greatly reduce asymmetric information.

To realize the goal mentioned in the last section, this thesis' has explored one approach: Create a system that facilitates open information flows, allowing entities to interact and store data with an emphasis on security.

9.2.2 Redundancy at Every Level

When creating a decentralized application, it is vital that *at no point* a third party is introduced. A third party managing *any* part of the system centralizes operation. By making the application completely dependent on a centralized third party, most benefits acquired by employing a blockchain vanishes.

Infura

Infura (subsection 6.2.11) provides a gateway to the Ethereum network, which is utilized in this PoC. While simplifying the set-up process, it spawns a centralized layer in applications utilizing it. Infura accounts for a substantial amount of value transfer as well as regular information requests [Pie18]. While being a fantastic tool for testing, allowing developers to deploy their dApps quickly, it is a centralizing hazard. By discontinuation of the Infura service, it would halt services for 35,000 dApps and 10 billion requests per day [inf18].

The *right* way of connecting to the Ethereum network, is through a full or light node, that the application itself is running. Currently, setting up a server and running the node yourself is the best approach. In the future, light nodes could reside on the device running the application, mitigating centralization.

Servers

A centralized database is introduced to relocate some work from the blockchain. If data that reside on the centralized database experience alterations, the application renders that change to the end user. However, the end user has no way of knowing what value was altered, or who altered it - only the fact that someone modified the data. This makes the solution tamper-evident, but not tamper-proof. This is a weakness compared to applications running fully on a blockchain.

Centralized storage as of today is called "location-based addressing" - when an actor wants to obtain a specific file, she utilizes the location of the server that has that file. That server responds with the specified file. IPFS is another form of off-chain storage structure. The system utilizes "content-based addressing", where the actor tells the computer what content to look for, rather than where it should look. The actor asks the computer to locate a hash "XYZ" of a particular file. When the actor receives the file, it can be re-hashed and verified against your local version of the hash, much alike the PoC created in this thesis. This p2p file system decentralizes storage, neglecting the need for several centralized databases. In a system that combines IPFS and a blockchain, the blockchain would store an immutable history while IPFS would store the immutable content.

The Blockchain

The Ethereum blockchain is considered decentralized, even though a certain number of miners control a vast amount of computational power, as mentioned in section 3.3.1. While the event of the Ethereum blockchain collapsing is implausible, safeguarding against such occurrences enhances the PoC robustness. If allowing the actors in the network to store fingerprints of information on several blockchains, one of them breaking down is an insignificant event. As of now, the Ethereum blockchain breaking down would cause the services provided by this application to be inaccessible.

9.2.3 Physical Component

The current PoC utilizes QR-codes as the physical counterpart. In a production version of this system, the physical elements should possess three features; tamper resistant, easy to use and cheap relative to the product.

- Tamper resistance: the component should not be removable, reproducible, transferable or mutable.
- Easy to use: End users should be able to authenticate with regular user equipment, e.g. a smart-phone.
- Cheap relative to the product: The cost of embedding physical elements in a product should be negligible compared to the value of that product. Enhancing the cost-benefit ratio is important for the system to be regarded as justifiable.

QR-codes excel in two out of three features - they are, however, not tamper-resistant [KLM$^+$10, p. 6]. Since its inception in 1948, RFID has been developed into a technology used every day [Lan05, p. 4]. From preventing theft of automobiles to entering buildings. With recent development in the 21st century, RFID-technology has become relatively cheap and easy to use, in the form of tags or stickers [Lan05, p. 4]. While QR-codes are cheaper than RFID, these stickers are tamper-evident [CA06], a significant advantage over QR-codes.

QR-codes are also unsuitable for perishable goods such as fish, fresh vegetables or meat. The system does not support tracking and authenticating such goods. The physical component requires transformation from static QR-codes to adhesive RFID chips.

9.2.4 Organizational Culture Limitations

During the product counterfeit background analysis, it became evident that many factors drive counterfeit product markets. These drivers can be split into three areas; the drivers the proposed system alleviates, the drivers it can alleviate with future endeavours and the drivers which it *cannot* reduce. In the two first areas, several issues stem from organizational culture.

Cost Versus Value Creation

Considering that the system would be more expensive than current anti-counterfeit measures, it has to create tangible value for those opting to use it. Despite the introduction of off-chain storage, the system is expensive when operated at a large scale, a significant weakness. It is also important to note that the cost of securing one product could be deemed too large compared to the value of that product.

Manufacturers and Subcontractors

If a manufacturer utilizes the proposed system, the current system can not prevent them from outsourcing this production to subcontractors. When intermediary parts are delivered, they can embed physical elements to the parts, which then will be secured by the system. The system is thus not resilient against counterfeiting prior to the product entering the legitimate supply chain.

Illicit Supply Chains

The sheer amount of counterfeit goods transported make it increasingly difficult for customs to inspect all containers. This solution does not address forged merchandise supplied by illicit logistic networks. The system is unfit to address consumer demand for counterfeit goods supplied by illicit logistic networks. Robust institutional frameworks that punish these illicit networks is a far better approach for this matter.

Silo Mentality

In order for this system to fully flourish, a vast amount of cognitive processes of central actors in the supply chain space need to change. Every intermediary that handle products along their life-cycle need to accept the system. Thus, addressing the silo mentality is more than providing technological advancements. It is about proving that the direct costs of utilizing this system are lower than the indirect losses of counterfeit goods.

Chapter 10 Conclusion

This chapter presents the conclusions of this thesis. The background analysis and implementation of the application have revealed the most pressing issues. Furthermore, ways to overcome the limitations that the proposed system carry are described in future work.

10.1 Research Question One

What technical modifications are essential to make current blockchain solutions viable for businesses?

- Enhance smart contract security
- Increase scalability issues regarding storage, responsiveness and throughput
- Choosing an applicable consensus algorithm
- Increase possibilities of privacy regarding storage and computation
- Stabilizing computation cost

Currently, all issues are addressed by the community.

RQ1.1

The issue regarding smart contract security is both about providing strict guidelines, and technological advancements. A combination of the three identified approaches appears as a sound approach. Although a combination slows down the process of building smart contracts, the compromise on speed improves security, which is paramount for critical code.

RQ1.2

Saving large amounts of data on the blockchain is extremely expensive. Currently, off-chain storage is required. However, there are approaches such as IPFS that does not compromise on decentralization. If file availability can be guaranteed by introducing economic incentives, IPFS solves the storage issue.

Throughput is currently low, but recent developments show promise. Altering the core protocol currently appear as the best solution.

As discussed, reducing latency to the level of centralized services seem improbable. Although layer-2 solutions provide latency at these levels, they compromise on decentralization with the re-introduction of intermediaries. They are also only applicable to recurring payments. If the application demand responsiveness regarding *writes*, a blockchain is not the appropriate instrument.

RQ1.3

Proof of work is the most thoroughly tested and appears as the most secure. However, for blockchains running smart contracts, proof of stake appear most promising. It increases throughput and can decrease latency.

10.2 Research Question Two

How can a system based on blockchain technology diminish counterfeit products from entering supply chains and markets?

This thesis concludes that blockchain technology can provide much sought after transparency to the supply chain. The decentralized application proves that properties such as immutability and transparency provided by the blockchain help facilitate information distribution, both between intermediaries and to the end user. This is done by scanning embedded physical elements that reside on products and digitize the recorded data. This open data is made tamper-evident by harnessing the immutable storage provided by the blockchain. Open data induce transparency, which helps excel information auditability. Enabling customers to examine a digital paper trail created on a network based on distributed trust, where they can fully trust all information shown, is an essential measure to combat product counterfeit.

As discussed in subsection 9.2.4, this solution is not able to rapidly change organizational cultures by *only* providing technological advancements. The proposed system needs to be deployed at several organizations throughout the supply chain at once. A comprehensive solution to the product counterfeit is thus a great deal more than just providing technology facilitating transparency and immutability. It is about disrupting the silo mentality that defines many entities in the supply chain ecosystem.

The proposed system carries some limitations that require improvements. The current database structure, physical elements and the code are all subject to change. How to improve on said limitations is described below.

10.3 Future Work

Migrate From Google Firebase to IPFS

Identified as the most important modification is the migration from Google Firebase to the IPFS structure. By relying on only one centralized server as off-chain storage, the application discard tamper-proof for tamper-evident - defined as a significant weakness in chapter 9. However, no radical code modifications are needed to solve this issue.

Substitute QR-codes with RFID-stickers

Another important measure is replacing current QR-codes with RFID-stickers. Tamper-evident RFID-stickers provide significant advantages over QR-codes. Although cost would increase, the declining price of RFID technology provides possibilities of securing a wide range of products.

As discussed in chapter 9, the introduction of passive RFID-stickers that can communicate would enable the system to secure perishable goods as well. Sensors situated in appropriate length from these goods can interact with the stickers, and provide a vast range of data to all actors in the network, such as temperature, humidity and illumination.

Code Optimization

The current code, including front-end, middleware and the smart contract can be optimized.

The smart contract that runs on the Ethereum network is currently the only contract handling information delivered by the front end. The smart contracts should map to a wide variety of products. E.g, a car company adopting the system require other parameters than a luxury clothing company. A mechanism to spawn smart contracts based on specifications that map to the needs of those adopting the system is possible.

The limitations the current smart contract experience is strict in order to minimize GAS-prices for those utilizing it. By having smart contracts with few computations, the cost of running the contract is low. If the functionality is expanded, more research into cost-effective smart contract coding should be conducted.

Addressing Organizational Culture

All future work mentioned above is strictly technical. However, one of the most critical issues going forward is the organizational culture in the space. The indirect nature of the cost of counterfeit products - loss of goodwill and trademark dilution, is complicated to address. It requires tremendous effort to provide sufficient data to show that the *direct* expense of running the proposed system is more cost-efficient compared to current counterfeit countermeasures.

References

[B$^+$13] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 2013.

[BBB19] Mahdi Zamani Benedikt Bünz1, Shashank Agrawal and Dan Boneh. Zether: Towards privacy in a smart contract world. 2019.

[BRO17] PAUL BROWN. State of the union: npm. January 2017. [Accessed 16/4/19].

[BTC19] BTC.com. Mining pool distribution. https://btc.com/stats/pool, May 2019. [Accessed 15/5/2019].

[But15] Vitalik Buterin. On public and private blockchains. 2015.

[But18] Vitalik Buterin. Explanation of daicos. *Ethereum Research*, 2018.

[CA06] Kevin Girard Conwell and Matt Adams. Tamper evident smart label with rf transponder, August 22 2006. US Patent 7,095,324.

[Caf15] Grace Caffyn. Everledger brings blockchain tech to fight against diamond theft. August 2015.

[Com] React Native Community. Create react native app. https://github.com/react-community/create-react-native-app. [Accessed 5/2/2019].

[CZ13] Peggy Chaudry and Alan Zimmermann. *Protecting Your Intellectual Property Rights*, chapter The Global Growth of Counterfeit Trade. 2013.

[Dre17] Daniel Drescher. *Blockchain basics.* Springer, 2017.

[EXP] Expo. https://expo.io/. [Accessed 5/2/2019].

[FMM08] Stanley E Fawcett, Gregory M Magnan, and Matthew W McCarter. Benefits, barriers, and bridges to effective supply chain management. *Supply Chain Management: An International Journal*, 13(1):35–48, 2008.

[FN16] C. K. Frantz and M. Nowostawski. From institutions to code: Towards automated generation of smart contracts. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 210–215, September 2016.

[Fou18] VeChain Foundation. Development plan and whitepaper. May 2018.

[FS18a] John Fernie and Leigh Sparks. *Logistics and retail management: emerging issues and new challenges in the retail supply chain.* Kogan page publishers, 2018.

[FS18b] Kristoffer Francisco and David Swanson. The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency. *Logistics*, 2(1):2, 2018.

[Ger17] David Gerard. *Attack of the 50 foot blockchain: Bitcoin, blockchain, Ethereum & smart contracts*, chapter Smart Contracts, Stupid Humans. David Gerard, July 2017.

[Gro17] Tom Groenfeldt. Ibm and maersk apply blockchain to container shipping. *URL: https://www. forbes. com/sites/tomgroenfeldt/2017/03/05/ibm-and-maersk-apply-blockchain-to-container-shipping*, 2017.

[GZ15] Alex 'Sandy' Pentland Guy Zyskind, Oz Nathan. Enigma: Decentralized computation platform with guaranteed privacy. 2015.

[Hol07] Gerard J. Holzmann. The power of ten – rules for developing safety critical code. *NASA/JPL Laboratory for Reliable Software*, January 2007.

[IA82] Yuji Ijiri and American Accounting Association. *Triple-entry bookkeeping and income momentum.* Sarasota, Fla : American Accounting Association, 1982.

[ICO] Funds raised in 2017. https://www.icodata.io/stats/2017. [Accessed 12/3/2019].

[inf18] Ethereum's node infrastructure provider infura handling 10 billion requests per day. https://www.trustnodes.com/2018/07/25/ethereums-node-infrastructure-provider-infura-handling-10-billion-requests-per-day, 2018. [Accessed 15/4/2019].

[Kea86] Brian J. Kearney. The trademark counterfeiting act of 1984: a sensible legislative response to the ills of commercial counterfeiting. *Fordham Urban Law Journal 14*, 1986.

[KLM+10] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar R. Weippl. QR code security. In *MoMM'2010 - The Eighth International Conference on Advances in Mobile Computing and Multimedia, Paris, France*, pages 430–435, November 2010.

[Lan05] J. Landt. The history of rfid. *IEEE Potentials*, 24(4):8–11, Oct 2005.

[Lee11] Charles Lee. Litecoin white paper, 2011.

[Mal18] Nikita Malik. Terror in the dark. *The Henry Jackson Society*, August 2018.

[Mol96] P. C. Molan. *Authenticity of honey*, pages 259–303. Springer US, Boston, MA, 1996.

[Nad18] Michael Nadeau. What is cryptojacking? how to prevent, detect, and recover from it. 2018. [Accessed 13/5/19].

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[NBF+16] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies: A comprehensive introduction.* Princeton University Press, 2016.

[oPH17] International Association of Ports and Harbours. World container traffic data. 2017.

[PD16] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016.

[Pie18] Maurycy Pietrzak. In 2017, infura relayed over 7 million eth. https://blog.infura.io/in-2017-infura-relayed-over-7-million-eth-over-8-billion-at-todays-prices-6c6cc7f73dd5, 2018. [Accessed 15/4/2019].

[PS16] Stéphane Jacobzone Michal Kazimierczak Nathan Wajsman Piotr Stryszowski, Florence Mouradian. Trade in counterfeit and pirated goods. pages 5 – 6, May 2016.

[RLD+17] B Rakic, T Levak, Z Drev, S Savic, and A Veljkovic. First purpose built protocol for supply chains based on blockchain. *OriginTrail, Ljubljana, Slovenia, Tech. Rep*, 1, 2017.

[RN] Raiden network. https://raiden.network/faq.html. [Accessed 25/4/2019].

[SB15] Mark Stevenson and Jerry Busby. An exploratory analysis of counterfeiting strategies: Towards counterfeit-resilient supply chains. *International Journal of Operations & Production Management*, 35(1):110–144, 2015.

[SI] Semantic versioning 2.0.0. https://semver.org/#spec-item-4. [Accessed 12/4/2019].

[Ste15] Rod Stephens. *Beginning Software Engineering.* 2015.

[Swa15] Melanie Swan. *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[Sza96] Nick Szabo. Smart contracts: Building blocks for digital markets. 1996.

[Sza05] Nick Szabo. Bit gold. http://unenumerated.blogspot.com/2005/12/bit-gold.html, April 2005. [Accessed 2/6/2019].

[TCK16] Patentstyret Toll Customs and Kulturdepartementet. http://www.velgekte.no/shopping/shopping-i-utlandet/europa/. June 2016. [Accessed 12/11/2018].

[Tho11] Ted Thornhill. The fake apple store in china so convincing that even its staff are fooled. 2011. [Accessed 15/4/19].

[Tim18] The Strait Times. Xi denounces vaccine safety scandal as vile and shocking. 2018. [Accessed 28/4/19].

[TRF] Truffle framework. https://truffleframework.com. [Accessed 6/2/2019].

[Tua16] Stephen Tual. No dao funds at risk following the ethereum smart contract 'recursive call' bug discovery. https://blog.slock.it/no-dao-funds-at-risk-following-the-ethereum-smart-contract-recursive-call-bug-discovery, 2016. [Accessed 10/5/19].

Appendix A

Appendix

Source code for the project is found in this GitHub repository:

https://github.com/glendur/AuthenticateProduct

Sigurd Glende

Blockchain: Business Potentials

NTNU
Norwegian University of
Science and Technology